UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

Análisis, Diseño e Implementación de un Esquema de Protección para las Comunicaciones de un Drone

Alexandre Erofeev Vasiltsov
Tutor: Oscar Delgado Mohatar
Ponente: Álvaro Ortigosa Juárez

MAYO 2017

Análisis, Diseño e Implementación de un Esquema de Protección para las Comunicaciones de un Drone

AUTOR: Alexandre Erofeev Vasiltsov
TUTOR: Oscar Delgado Mohatar

Departamento de Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Junio de 2017

Resumen (castellano)

En los últimos años, los drones de uso civil se han hecho cada vez más populares. Es más común su uso no solo para el entretenimiento personal, sino que numerosas empresas lo utilizan con diversos fines, como la grabación de vídeo, entrega de productos (por ejemplo Amazon Prime Air), vigilancia, control de incendios, así como muchos otros.

Sin embargo, el avance tecnológico nunca debe llevar a dejar de lado otros aspectos importantes, entre ellos la seguridad. La innovación que se ha llevado a cabo en el mundo de los drones en los últimos años no se ha dado de forma igual en lo que respecta a la seguridad de éstos. Y hoy en día esto supone uno de los mayores retos (si no el mayor) relacionados con esta industria.

La gran mayoría de los drones han dejado completamente de lado las cuestiones relacionadas con la seguridad, y casi cualquier dron es vulnerable a numerosos ataques que pueden suponer daños para la reputación del fabricante, para el usuario o para la empresa. Es por ello que debe haber una concienciación al respecto, y un trabajo duro para solucionar estos fallos de cara al futuro.

El objetivo de este trabajo consiste precisamente en analizar en profundidad las vulnerabilidades que existen en la comunicación entre un dron y el cliente que lo maneja, en describir los ataques más comunes que pueden ser realizados, y por último, en proponer una o varias soluciones a estos fallos de seguridad.

En una primera sección se analizarán las vulnerabilidades del dron, haciendo uso de diversas técnicas, como el escaneo de puertos para descubrir puertos abiertos que ejecuten diversos servicios, o la captura de tráfico, para comprender como se envían órdenes al dron.

En la segunda sección, una vez descubiertas las vulnerabilidades, se describirán los ataques más comunes que pueden ser realizados. Entre estos ataques se encuentran el envío de órdenes falsas al dron, o el acceso a servicios como Telnet o FTP.

Finalmente, en una última sección, se propondrán y describirán una serie de soluciones a estas vulnerabilidades.

Abstract (English)

Over the past few years, the popularity of drones used for civil purposes has notably increased. They are not only used for personal entertainment, but lots of companies have found them useful in order to help them doing different tasks, such as video recording, product delivery (everybody have heard about Amazon Prime Air), surveillance, fire control… the list can be endless.

However, no matter how far technological progress has gone, we cannot forget about security issues. While lots of improvements have been done in this area, security hasn't been one of those improvements. Nowadays, security is one of the most important challenges (if not even the most important) drone research area has to deal with.

Most part of the drones haven't even bother about security related issues, and almost every drone is vulnerable to a wide variety of attacks which can produce a great damage to the drone's company reputation, the user, or the company. That's the reason why drone producing companies and drone users have to be concerned about that, and why hard work must be done to solve these problems.

The main purpose of this project is to analyze every vulnerability that exists in the communication between the drone and the user who controls it, to describe the most common attacks that can be done, and finally, to propose some solutions to make the drone communication more secure.

In the first section, drone's vulnerabilities will be analyzed, making use of various techniques, such as port scanning in order to discover open ports where some well-known services are running, or packet sniffing, to understand how user sends orders to the drone.

In the second section, once all vulnerabilities are already discovered, most common attacks that can be done will be described. These attacks include such attacks sending fake orders to the drone, or getting Telnet and FTP access.

Finally, in the last section, some countermeasures will be proposed in order to make the communications between the drone and the user safer.

Palabras clave (castellano)

Dron, dron de uso civil, dron incendios, seguridad, comunicaciones seguras, comprometer dron.

Keywords (inglés)

Drone, civilian use drone, fire control drone, security, secured communications, drone hacking, UAV

Agradecimientos

Quisiera agradecer en primer lugar a mi familia, que siempre me ha proporcionado el apoyo necesario para que siga adelante y gracias a ella he llegado hasta aquí. Asimismo quisiera agradecer al tutor por la ayuda dada a la hora de hacer el trabajo.

INDICE DE CONTENIDOS

INDICE DE FIGURAS

INDICE DE TABLAS

1 Introducción

1.1 Motivación

En los últimos años se han producido numerosos avances en el campo de los drones de uso civil y su popularidad ha aumentado notablemente. Ya no son solo usuarios quienes con el único objetivo de entretenerse hacen uso de estos drones, sino que diversos organismos han encontrado éstos drones útiles para la realización de diversas tareas que antes eran irrealizables o tenían un coste mayor, tales como la retransmisión de vídeo en directo, la vigilancia, el envío de pedidos, etc. Sin embargo, sorprende que al mismo tiempo que se haya avanzado tanto en el desarrollo de drones para uso civil, se haya obviado una cuestión que no es en absoluto trivial: la seguridad de las comunicaciones entre un dron y el cliente que lo maneja.

Por estas razones en este trabajo se ha llevado a cabo un proceso de análisis de vulnerabilidades, ataques a las mismas y propuestas de contramedidas para los problemas encontrados. El proyecto, ha sido llevado a cabo sobre el dron **Parrot A.R. Drone 2.0**, pues se ha considerado a este dron como una muestra representativa de los drones más populares en la actualidad.

Este trabajo no es el primero ni el único que se lleva a cabo acerca de la seguridad de las comunicaciones entre este dron y el cliente que lo maneja. Se han llevado varias investigaciones acerca del tema, entre las cuales habría que destacar la investigación realizada por Pleban *et al* de la Universidad de Bradenburgo [1] y la realizada por Mark Szabo [2], las cuales han servido de referencia e influencia en la realización de este trabajo.

Posteriormente, en el apartado '**Estado del arte**' se describirá con mayor detalle la forma de comunicarse del dron escogido, y se comentarán con algo de detalle algunos de los trabajos realizados acerca de la seguridad de las comunicaciones del Parrot A.R. Drone 2.0, proporcionando un enfoque más global sobre la problemática a tratar.

1.2 Objetivos

Los objetivos de este trabajo son, a grandes rasgos:

- **Analizar** en detalle **el funcionamiento de las comunicaciones** entre un dron y la persona que lo maneja.
- **Analizar** las potenciales **vulnerabilidades** y **fallos de seguridad**.
- Una vez descubiertas las vulnerabilidades, **describir** los posibles **ataques** que se pueden realizar en una situación real.
- Finalmente, describir una serie de **contramedidas** para protegerse de estos ataques y que en definitiva puedan hacer la comunicación entre el dron y el cliente más segura.

1.3 Organización de la memoria

La memoria consta de los siguientes capítulos:

- **Estado del arte**. Este capítulo se podría dividir a su vez en dos partes.

- **Estudio de los drones más vendidos clasificados por precio**. Se realiza un estudio de algunos drones representativos del mercado, en distintos rangos de precios, y se cita su precio, algunas características, y la forma en la que se comunica con el cliente.
- **Trabajos realizados sobre el tema**. Se describen algunos trabajos ya existentes que guardan relación con la seguridad del dron AR Drone 2.0 y en las comunicaciones entre el dron y el cliente.

- **Descripción y análisis**. Este capítulo también se puede dividir en varias partes.

  - **Especificaciones técnicas del dron**. Aquí se citan las características técnicas del dron AR Drone 2.0.
  - **Funcionamiento del dron**. En este capítulo se describe cómo se maneja el dron desde la aplicación oficial.
- **Análisis de vulnerabilidades**. En esta sección se describen los pasos seguidos para descubrir las vulnerabilidades del dron.
- **Vulnerabilidades del dron que pueden ser explotadas**. Una vez descubiertas las vulnerabilidades tras seguir los pasos indicados en el apartado anterior, éstas se describirán en este apartado.

  - **Desarrollo. Ataques**. En esta sección se describirán los ataques que pueden realizarse explotando las vulnerabilidades descubiertas en el apartado anterior.
  - **Contramedidas**. En este capítulo se proponen y explican algunas medidas para tapar los agujeros de seguridad que existen en la comunicación entre el dron y el cliente.
  - **Conclusión**. Por último, aquí se mencionan las conclusiones finales y algunas propuestas de cara a un trabajo futuro.

2 Estado del arte

En este capítulo se describe el estado de la seguridad de las comunicaciones entre el dron y el cliente. En la primera parte, se escogen los drones más populares del mercado y se describen algunas de sus características y su forma de comunicarse. En la segunda parte, se describen los trabajos ya realizados acerca de la seguridad del dron AR Drone 2.0.

2.1 Estudio de los drones más vendidos clasificados por precio

En este apartado se llevará a cabo un estudio de la forma de comunicarse de los drones más populares, organizándolos por su precio de venta en Amazon en base a distintos rangos: > 500, 200 – 500, 50 – 200 y 20 – 50 €. [9]

2.1.1 Drones con un precio mayor que 500 euros

Nombre y precio	Algunas características	Forma de comunicarse
Parrot PF726203 - Drone Bebop 2 Precio: 628,26 €	Diseño ligero, 25 minutos de vuelo, cámara de 14 MP y grabación de vídeo 1080p HD, GPS incorporado y posibilidad de planear rutas de vuelo, etc. [6]	El cliente se conecta a una red Wi-Fi abierta y controla el dron a través del móvil, haciendo uso de una aplicación. [7]
DJI Phantom 3 Pro Precio: 1.159,02 €	Cámara 1080p/12MP, vuelo estacionario, batería inteligente, etc. [8]	Se controla desde un mando en una frecuencia de 2.4GHz. El teléfono móvil o la Tablet se conectan al mando por USB, el cual se encarga de hacer de enlace entre el terminal y el dron. [10]
3DR - SA15A Precio: 1.106,65 €	16 minutos de vuelo, cámara Sony R10C de 20MP, rango de comunicación de un km, etc. [11]	Se controla desde un mando en una frecuencia de 2.4GHz. El *streaming* de vídeo se puede ver en una aplicación que el usuario se descarga en el dispositivo móvil, y la comunicación se produce dentro de una red Wi-Fi segura. [11]

Tabla 2.1: Drones con un precio mayor que 500 euros

2.1.2 Drones con un precio entre 200 y 500 euros

Nombre y precio	Algunas características	Forma de comunicarse
NincoAir STRATUS WIFI GPS Precio: 236,04 €	Cámara HD, GPS, Batería Li-Po 7,4V 2000mAh [12]	El control del dron se realiza por medio de un mando que se comunica con el dron en una frecuencia de 2.4GHz, la retransmisión de vídeo se realiza a través de una red Wi-Fi. [12]
Parrot - Drone Bebop (PF722000) Precio: 289,57 €	Ultraligero, Cámara 1080p/14 MP, 11 minutos de uso, etc. [13]	El cliente se conecta a una red Wi-Fi abierta y controla el dron a través del móvil, haciendo uso de una aplicación (similar al Bebop 2).
Hubsan H501S Precio: 279,00 €	GPS, Cámara 1080p, 20 minutos de vuelo [14]	El cliente utiliza un mando para manejar el dron y la conexión es en una frecuencia de radio que puede ajustarse en el mando. [15]

Tabla 2.2: Drones con un precio entre 200 y 500 euros

2.1.3 Drones con un precio entre 50 y 200 euros

Nombre	Algunas características	Forma de comunicarse
Syma X5C-1 Precio: 59,99 €	5-8 min de vuelo, cámara HD, Micro SD de 2Gb, ligero. [16]	Control por radio sobre una frecuencia de 2.4GHz a través de un mando. [16]
GoolRC T5G Precio: 64,60 €	Cámara HD 2.0MP, 4-5 minutos de vuelo, ligero. [17]	Control por medio de radiofrecuencia haciendo uso de un mando. [17]
OCDAY M65500 Precio: 52,99 €	Cámara de 0.2MP, resistente al viento, puede moverse por el suelo además de por el aire. [18]	Control mediante mando a través de radiofrecuencia, *streaming* de vídeo retransmitido por Wi-Fi. [18]

Tabla 2.3: Drones con un precio entre 50 y 200 euros

2.1.4 Drones con un precio entre 20 y 50 euros

Nombre	Algunas características	Forma de comunicarse
KYG JJRC H36 Precio: 22,99 €	Ligero, simple, sin cámara [19]	Control mediante mando a través de radiofrecuencia de 2,4GHz. [19]
EACHINE E010 Mini UFO Precio: 20,99 €	5 minutos de vuelo, 30 metros de distancia de vuelo, simple [20]	Control mediante mando a través de radiofrecuencia de 2,4GHz. [20]
RC Quadcopter Precio: 35,99 €	Cámara de 0,3 MP HD, 7 minutos de vuelo, simple [21]	Control mediante mando a través de radiofrecuencia de 2,4GHz. Los vídeos realizados por la cámara se graban en un USB. [21]

Tabla 2.4: Drones con un precio entre 20 y 50 euros

2.1.5 Conclusión del estudio

Tras analizar las formas de conectarse de los distintos drones, llegamos a la conclusión de que los drones son en su mayoría controlados mediante un mando a través de una señal de radio, o mediante una aplicación móvil, donde habitualmente el cliente se conecta a una Wi-Fi (muchas veces abierta) del dron donde las órdenes se envían mediante protocolos específicos. Algunas veces la interacción entre el cliente y el dron se produce de manera mixta: mientras las órdenes de control se transmiten por radiofrecuencia, el vídeo se retransmite mediante una conexión Wi-Fi.

2.2 Trabajos realizados sobre el tema

A pesar de la importancia de la cuestión de la seguridad en las comunicaciones entre el dron y el cliente, no se han realizado abundantes investigaciones. Sin embargo, aunque se hayan realizado pocos estudios, sobre este dron en concreto existen algunos trabajos que merece la pena leer y comentar y que sin duda han supuesto una influencia en este proyecto.

2.2.1 Trabajo realizado por investigadores de la universidad de Bradenburgo

Este trabajo [1], llevado a cabo sobre el mismo dron que se ha utilizado en este proyecto, posiblemente sea uno de los estudios más elaborados que existen hasta la fecha acerca de las comunicaciones entre dron y cliente.

En este estudio se tocan varios aspectos en lo que concierne al dron. Tras una introducción, analiza en detalle las características del dron (tanto de hardware como de firmware/software) así como la manera en la que un cliente puede manejarlo. Continúa describiendo algunos de los usos originales que se han hecho del dron y que posibilidades puede ofrecer más allá de lo que puede ofrecer cualquier dron. Posteriormente, analiza las vulnerabilidades y fallos de seguridad que tiene el dron. Y esto es lo que nos interesa a nosotros.

Los autores se conectan desde su ordenador a la red Wi-Fi que genera el dron (tal y como se conectaría alguien para manejarlo desde su terminal). A continuación, hacen un escáner de puertos utilizando **nmap**, descubren los puertos abiertos, y describen su uso en la siguiente tabla:

Puerto	Uso
21 (TCP)	Servidor FTP que contiene fotos y videos realizados por el dron.
23 (TCP)	Servidor telnet.
5551 (TCP)	Acceso por FTP a la carpeta update con el objetivo de actualizar el firmware.
5553 (TCP)	Transmisión de vídeo H264-720p si la opción de grabar está activada.
5554 (UDP)	NAVDATA: por este puerto se envían los datos de telemetría (estado, velocidad).
5555 (TCP)	Stream del vídeo visible cuando el usuario maneja el dron.
5556 (UDP)	ATCMD: El dron está controlado por comandos AT que se envían al dron de manera periódica (aproximadamente 30 cmds/segundo)
5559 (TCP)	CONTROL: por este puerto se envían datos de configuración

Tabla 2.5: Puertos abiertos del dron AR Drone 2.0 y uso

A continuación los autores del artículo examinan y describen más a fondo las funciones que tienen algunos puertos:

- **FTP**. La conexión al servicio FTP no está protegida por contraseña, lo que permite que cualquier usuario pueda acceder de manera anónima al subdirectorio **/data/video** del dron. Después de conectar un dispositivo USB al dron, este es montado en **/data/video/usb/**. Esto permite acceso al dispositivo conectado de manera que el atacante pueda acceder a archivos confidenciales o introducir en el dispositivo archivos maliciosos.

- **Telnet**. Uno puede iniciar una sesión telnet con el dron y accederá como **root** sin necesidad de introducir contraseña. A partir de aquí las posibilidades del atacante son inmensas, pues tiene el control absoluto de la máquina. Los autores también han elaborado una lista de los scripts en bash que pueden interesar al atacante:

Dirección	Uso
/bin/check_update.sh	Script de actualización.
/bin/init_gpios.sh	Inicialización de los puertos GPIO.
/bin/mount_usb.sh	Monta el dispositivo USB.
/bin/pairing_setup.sh	Asocia el dispositivo móvil con el dron.
/bin/parallel-stream.sh	Script de *streaming* de vídeo.
/bin/reset_config.sh	Resetea el archivo config.ini.
/bin/umount_usb.sh	Desmonta el dispositivo USB.
/bin/Wifi_setup.sh	Inicia la conexión Wi-Fi y otros servicios.

/sbin/udevd.sh	Inicia el servicio udevd con el lanzador de udevd_init.
/lib/udev/rndis.sh	Servicio-gancho llamado por udhcpc en eventos relacionados con la interfaz rndis.
/usr/sbin/loadAR6000.sh	Configuración adicional de Wi-Fi
/etc/inetd.conf	Servidor FTP (/update y /data/video)
/etc/udhcpd.conf	Configuración DHCP para la red Wi-Fi
/data/config.ini	Principal fichero de configuración.

Tabla 2.6: Scripts en bash del dron que pueden interesar al atacante

- **Comandos AT**. El autor describe como se mandan los comandos desde el terminal para manejar al dron. Estos comandos son mandados utilizando UDP y para controlar el orden de llegada de comandos se utilizan números de secuencia. Un atacante podrá enviar comandos simplemente suplantando la IP del cliente y o bien enviando un comando con un número de secuencia mayor al del último comando enviado o bien enviando el número de secuencia 1, lo cual reseteará el contador interno del dron.

Posteriormente se describen algunos escenarios de ataque posibles:

- **Telnet**. Tal y como se ha mencionado brevemente en la descripción de los servicios, aquí se vuelve a repetir que al poder autenticarte como root sin contraseña, cualquier ataque es posible, desde simplemente apagar el dron mientras está volando hasta algo más elaborado.

- **FTP**. Se vuelven a repetir los posibles ataques que se han mencionado en los anteriores apartados.

- **Puertos de control**. Se menciona que el atacante puede interceptar el stream de vídeo o enviar nuevos datos de configuración al dron.

- **Multiusuario**. El sistema Linux del dron es multiusuario, por lo que el atacante podría crear un nuevo usuario y ejecutar código desde ahí.

- **Combinación de ataques**. Aquí se menciona el SkyJack-Hack desarrollado por Samy Kamkar, el cual se comentará en un apartado posterior.

Después se menciona el sistema de seguridad que ha implementado el dron, el cual consiste en una asociación (en inglés *pairing*) del dron con el terminal. Una vez se ha asociado el dron con el terminal, todo el tráfico no autorizado es bloqueado.

Hacer que la comunicación sea segura

Finalmente el artículo propone como **solución** a la inseguridad reinante utilizar una **red Wi-Fi segura** (**WPA** o **WPA2**). Se explica cómo instalar la herramienta **wpa_supplicant** para un dispositivo cuyo set de instrucciones es ARM (como en este caso). Una vez instalada esta herramienta, el proceso consistiría en lo siguiente: se necesitarían tres dispositivos, uno de los cuales sería el encargado de generar la red Wi-Fi segura. Una vez generada esa red Wi-Fi segura, otro dispositivo se conectaría a la Wi-Fi insegura del dron por telnet y escribiría una serie de comandos para que se establezca una conexión entre el dron y el punto de acceso

seguro. A partir de aquí el terminal podrá establecer una conexión segura con el dron. Este proceso a priori puede resultar tedioso (sobre todo por la necesidad de un tercer dispositivo), aunque se podría automatizar. Sin embargo, hay que decir que tendría una ventaja adicional: la red Wi-Fi del tercer dispositivo podría estar conectada a Internet y eso aumentaría en buena medida las utilidades del dron.

2.2.2 Trabajo de Mark Szabo

Este trabajo [2], presentado en la conferencia de hacking ético de Budapest en Mayo de 2016, es otro estudio realizado acerca de la seguridad de la comunicación entre un dron y un terminal. Resulta interesante principalmente debido a que el autor no solo describe los fallos de seguridad encontrados, sino que describe como los ha encontrado, proporcionando capturas de pantalla y explicando los pasos que ha seguido. Asimismo, el autor proporciona el código que él ha utilizado para enviar comandos AT y poder controlar el dron. Pero sobre todo resulta interesante porque el autor explica como burlar el único sistema de seguridad que existe en la comunicación entre el dron y el terminal: el ***pairing***.

El *pairing*, tal y como se ha mencionado antes, consiste en una asociación entre el dron y el terminal. Cuando se emparejan el dron y el terminal, el dron registra la MAC del cliente, y utilizando reglas de iptables, permite el acceso del terminal asociado a todos los puertos abiertos y bloquea el acceso al resto de dispositivos de la red Wi-Fi a todos los puertos salvo el 21 (FTP), 2049 (NFS), y también permite que cualquier dispositivo haga uso del protocolo ICMP.

El autor describe como se podría burlar este sistema para enviar comandos para manejar el dron: no tiene mayor misterio que enviar un paquete con la MAC origen del terminal que está asociado al dron. Sin embargo, aun así no podremos acceder a telnet u otros servicios. Para este caso, el autor también tiene una solución: enviar un comando de configuración para desactivar esta medida de seguridad y así el atacante volverá a poder tener el control absoluto del dron. El autor nos proporciona un código desarrollado por él que automatiza esta tarea.

2.2.3 SkyJack

El conocido y controvertido hacker e investigador de seguridad Samy Kamkar, famoso por haber creado el gusano Samy [3], ha desarrollado un sistema capaz de detectar drones Parrot AR Drone 2.0, hacer que se desconecte el usuario anteriormente conectado y tomar el control absoluto del dron. De esta manera, se puede llegar a crear un ejército de drones zombie.

Para ello hace uso de una Raspberry Pi y hace uso de una tarjeta de red Alfa AWUS036H. La Raspberry Pi se alimenta mediante una batería USB externa. La Raspberry Pi pone la tarjeta de red en modo monitor y escanea las redes Wi-Fi cuya dirección MAC coincida con la MAC de un dron Parrot A.R Drone 2.0. Posteriormente realiza un ataque de deautenticación al usuario que esté conectado al dron haciendo uso de la conocida herramienta AirCrack. Posteriormente, para conectarse al dron y manejarlo, utiliza la librería node-ar-drone, la cual permite controlar el dron mediante Javascript y Node.JS.

En Youtube [4] se puede ver un vídeo donde Samy Kamkar muestra a su invento en acción y describe su funcionamiento a nivel de desarrollo. En su sitio web [5] podremos encontrar más información y un enlace a Github donde se encuentra el código del programa.

2.2.4 Conclusión

Los trabajos anteriormente descritos han sido los primeros en investigar en profundidad la seguridad del dron AR Drone 2.0 y han sentado las bases para una investigación más profunda. Al tener en disposición estos trabajos y al describirse en estos la comunicación entre el dron y el cliente con algo de detalle, y al explorar por encima algunas vulnerabilidades, ha sido relativamente sencillo saber por dónde empezar. Sin duda, estos trabajos, especialmente el realizado por investigadores de la Universidad de Bradenburgo, han tenido no poca influencia en la realización de este proyecto.

3 Descripción y análisis

En este capítulo, primero se describen las características técnicas del dron, a continuación se describe su funcionamiento, y finalmente se realiza un análisis de vulnerabilidades, las cuales, una vez conocidas, serán explotadas en el capítulo 4.

3.1 Especificaciones técnicas del dron

Algunas de las especificaciones técnicas del dron más importantes (para este trabajo) son las siguientes [23]:

- Procesador de **32 bits** con una frecuencia de **1 GHz** basado en **ARM Cortex A8**.
- **GPU PowerVR SGX530** con una frecuencia de **800 MHz**.
- Sistema operativo **Linux 2.6.32**.
- Memoria **RAM DDR2 1 GB a 200 MHz**.
- **USB 2.0** de alta velocidad para las extensiones
- Conexión: **Wi-Fi**.
- Cámara vertical: **QVGA 60 FPS** para medir la velocidad en vuelo
- **Cámara HD 720p 30 FPS**
- Formato fotos: **JPEG**

El resto de características técnicas pueden encontrarse en el **Anexo A**.

3.2 Funcionamiento del dron

El dron se maneja desde una aplicación llamada **AR.FreeFlight**, disponible en Google Play [24]. Lo primero que debemos hacer para manejar el dron (después de descargarnos la aplicación) es conectarnos a la red Wi-Fi del dron, cuyo ESSID es **ardrone2**. Una vez nos hayamos conectado, se nos abrirá una interfaz bastante intuitiva, en la cual tendremos un menú con varias opciones, entre ellas la de manejar el dron. La forma en la que se maneja el dron puede encontrarse explicada más detalladamente en el **Anexo B**.

3.3 Análisis de vulnerabilidades

En esta sección se describe el proceso seguido hasta encontrar todas las vulnerabilidades existentes en el dron.

3.3.1 Primeros pasos

Procedemos a continuación a describir los pasos que se llevaron a cabo para analizar la seguridad del dron y detectar posibles vulnerabilidades. El primer paso es conectarnos con nuestro ordenador a la red Wi-Fi del dron (todo este proceso ha sido realizado desde un ordenador con Ubuntu 14.04). El ESSID de la red Wi-Fi es **ardrone2** y es una red Wi-Fi abierta.
Ahora debemos averiguar la dirección IP del dron. Para ello, abrimos la terminal y tecleamos el comando **route**. Se nos mostrará la tabla de rutas IP. Nos fijamos en la dirección IP que

se sitúa debajo de "pasarela": la dirección **192.168.1.1** se corresponde (siempre) con la dirección IP del dron.

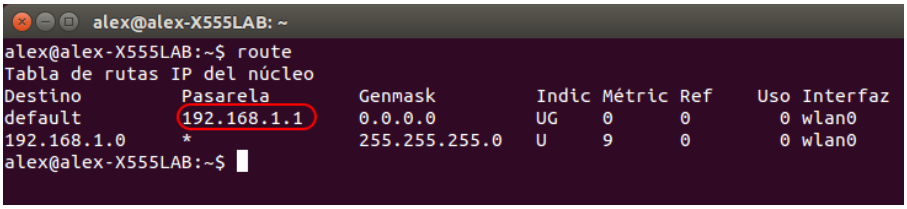


Figura 3.3.1: Comando route

Ahora que conocemos la dirección IP del dron podemos realizar un escáner de puertos y de esta manera averiguar que puertos tiene abiertos el dron y que servicios hacen uso de ellos. Para realizar el escáner de puertos nos valdremos de la herramienta **Nmap**. Ejecutamos el siguiente comando:

sudo nmap -sS -sU -PN -p 1-65535 192.168.1.1 -T5 –v

Y tras esperar un largo periodo de tiempo el programa nos muestra los resultados por pantalla.

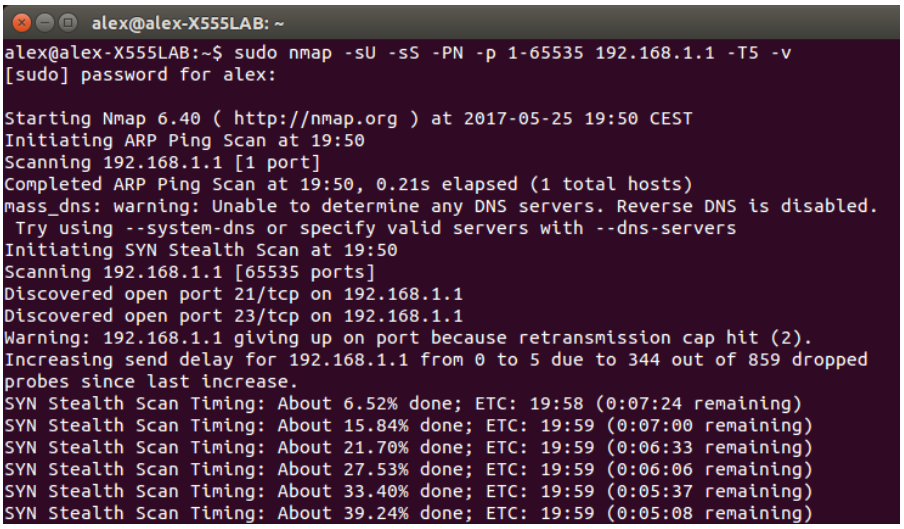


Figura 3.3.2: Resultados de nmap 1/2

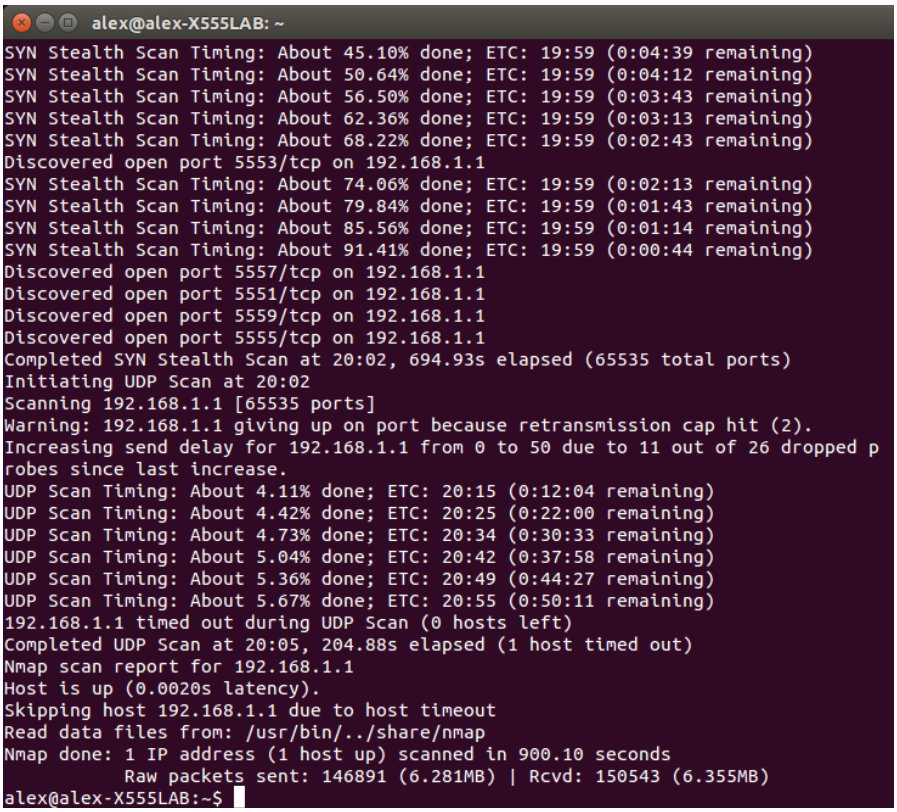


Figura 3.3.3: Resultados de nmap 2/2

Sin embargo, si nos fijamos en los puertos que hemos descubierto, nos damos cuenta de que hay algunos puertos UDP que sabemos que el dron usa (gracias a otros trabajos), y sin embargo Nmap no los ha mostrado en los resultados. Esto se debe a que estos puertos están abiertos, pero solo reciben información, no interactúan de ninguna manera con el cliente, por lo que no podemos saber si el puerto está abierto o no. Sin embargo, hay una forma de averiguarlo.
Nosotros hemos descubierto que el puerto 23 (telnet) está abierto. Nos conectamos a él (no se necesita contraseña):

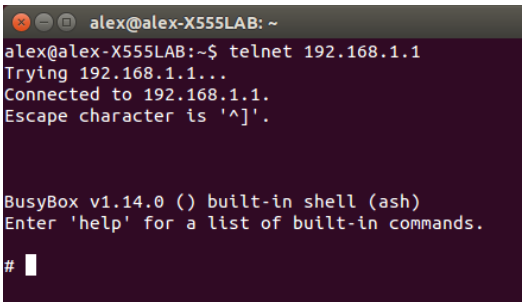


Figura 3.3.4: Conexión al dron a través de telnet (1)

Y ejecutamos el siguiente comando:

Netstat -a

Lo cual nos mostrará todos los puertos en los que el dron está escuchando:

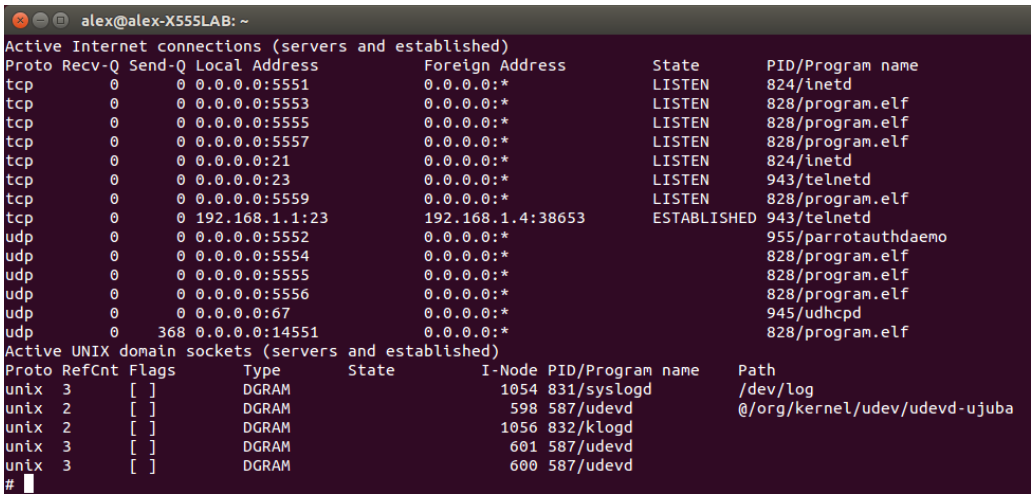


Figura 3.3.5: Resultados de ejecutar el comando netstat

3.3.2 Análisis de los servicios

A la hora de realizar el escaneo con nmap y tras ejecutar el comando netstat, hemos descubierto los siguientes puertos abiertos. A continuación se muestra una tabla donde se nombran y describen los servicios que corren sobre esos puertos:

Puerto	Nombre del servicio	Descripción
21 (TCP)	FTP (fotos y videos)	Aquí se guardan las fotos y los vídeos realizados por el dron.
23 (TCP)	Telnet	Servidor telnet, desde aquí se puede acceder a la Shell del sistema.
67 (UDP)	DHCP	Protocolo DHCP mediante el cual se asignan direcciones IP de forma dinámica.
5551 (TCP)	FTP (firmware)	A través de este puerto se realizan las actualizaciones de firmware.
5552 (UDP)	Handshake	El dron envía un paquete UDP multicast al que el cliente debe responder para ser reconocido por el dron.
5553 (TCP)	Transmisión de vídeo (H264-720p)	Transmisión de vídeo si la opción de grabar está activada.
5554 (UDP)	Datos de navegación.	A través de este puerto se envían datos de telemetría (estado, velocidad).
5555 (TCP)	Transmisión de vídeo	Transmisión de vídeo visible cuando el usuario maneja el dron.
5556 (UDP)	Envío de comandos al dron	A través de este puerto, se envían comandos que permiten manejar al dron.
5557 (TCP)	Ningún servicio.	Puerto activo sin ningún uso en particular.

5559 (TCP)	Control	A través de este puerto se envían datos de configuración.
14551 (UDP)	Desconocido	Desconocido

Tabla 3.1: Puertos abiertos del dron, nombre del servicio y descripción

3.3.3 Análisis del tráfico y suplantación de paquetes de orden de movimiento del dron

Una vez analizados los puertos y los servicios que se ejecutan en esos puertos, el siguiente paso consistirá en **analizar el tráfico** para ver **cómo se comunica el dron con el terminal**. Para ello, lo primero que tenemos que hacer es realizar un **ataque de envenenamiento de ARP** al cliente, y después mediante un ***sniffer*** analizar el tráfico que circula entre el dron y el cliente.
Antes de realizar el ataque deberemos ejecutar el siguiente comando:

echo 1 > /proc/sys/net/ipv4/ip_forward

Este fichero puede contener o un 0 o un 1. Si contiene un 0, el tráfico que ha pasado por nuestro ordenador pero va dirigido a otro host no sería reenviado. Si contiene un 1, es reenviado, lo cual es lo que nos interesa en este caso para analizar el flujo entre el dron y el cliente.

Ahora pasamos al ataque por envenenamiento de ARP. Para realizar este ataque, utilizaremos la herramienta **arpspoof** del paquete **dsniff**. Ejecutamos el siguiente comando:

sudo arpspoof -i wlan0 -t 192.168.1.2 192.168.1.1

Donde 192.168.1.1 es la IP del dron (siempre será la misma, tal y como dijimos antes) y 192.168.1.2 es la dirección del cliente. La dirección del cliente puede variar (dependiendo de quien se ha conectado antes). En este caso, realizaríamos un escáner con nmap para averiguar que equipos hay en la red, utilizando el siguiente comando:

nmap -sn 192.168.1.0/24

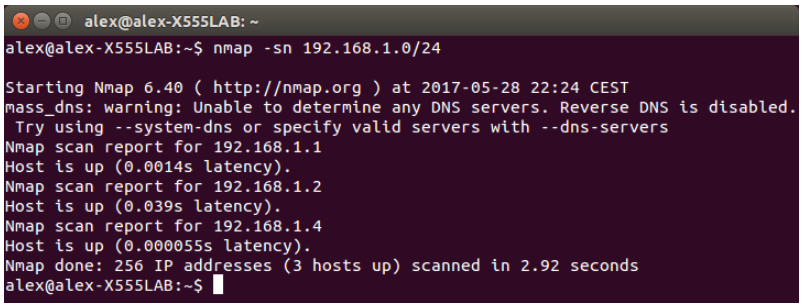


Figura 3.3.6: Escaneo de hosts

Y nos saldrán tres direcciones IP. Una es la del dron, otra la nuestra y otra del cliente. Ahora ejecutamos el comando **ifconfig** y nos fijamos en la interfaz wlan0 para conocer nuestra IP.

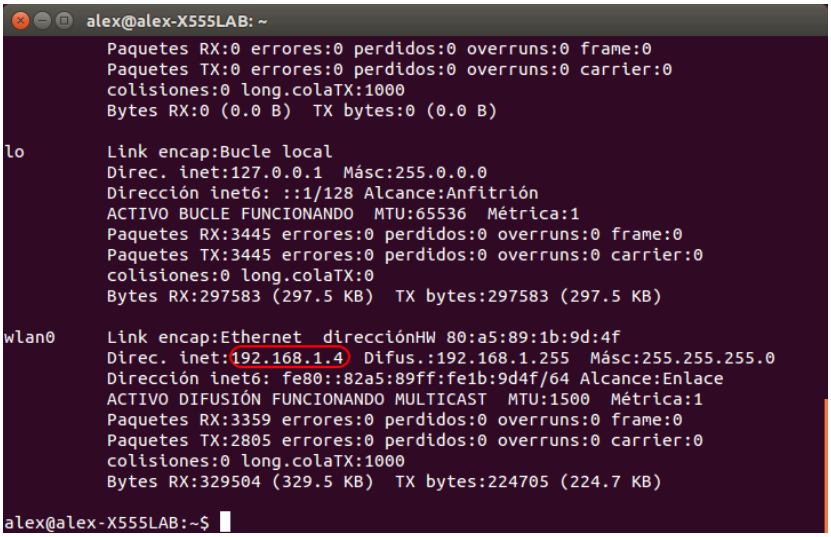


Figura 3.3.7: Resultados de ifconfig

La dirección que está dentro del recuadro (192.168.1.4) es nuestra IP. Por descarte, sabemos que la dirección 192.168.1.2 es la dirección IP del cliente.

Ahora que ya hemos realizado un ataque de envenenamiento de ARP abrimos **Wireshark** (el *sniffer* que hemos elegido) y procedemos a analizar el tráfico de la red. Aplicamos el siguiente filtro:

(ip.src == 192.168.1.2 or ip.dst ==192.168.1.2) and not icmp

Pues solo nos interesa lo que entra o sale de 192.168.1.2. Los paquetes ICMP abundan y no nos interesan, por lo que también los filtramos.

Ahora desde la aplicación del móvil entramos al control y observamos los paquetes que aparecen en Wireshark:

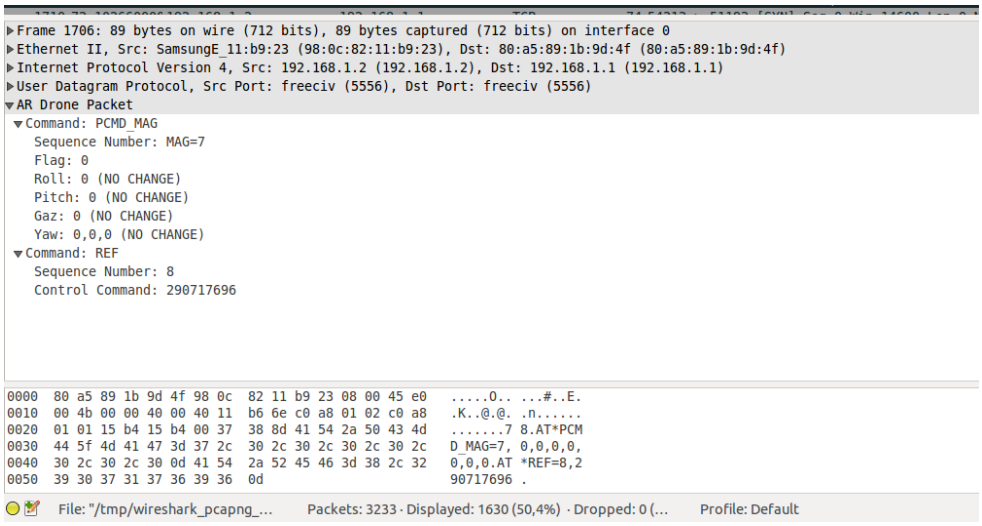


Figura 3.3.8: Comando AT

Podemos observar, entre otros, como el cliente se comunica con el dron por medio de **comandos AT**, y su forma y contenido. En el apartado siguiente se describirán en detalle las vulnerabilidades relacionadas con esta forma de comunicarse, así como otras tantas más.

3.4 Vulnerabilidades del dron que pueden ser explotadas

En esta sección se comentarán con algo de detalle las vulnerabilidades que han sido encontradas y que un atacante podría estar interesado en explotar. Una vez descritas estas vulnerabilidades, en el apartado ataques se describirán los ataques que un atacante en una situación real podría estar interesado en llevar a cabo.

3.4.1 Suplantación de órdenes de manejo del dron

En la figura 3.3.8 podemos ver como el cliente se comunica con el dron por medio de comandos AT, incluso antes de siquiera haber empezado a manejar el dron. Los distintos comandos AT y su funcionamiento se pueden ver aquí [22]. Cada instrucción AT contiene un número de secuencia. El primer paquete que se envía, se envía con un número de secuencia generado al azar. El dron solo ejecutará aquellas instrucciones con un número de secuencia mayor que el anterior. En el artículo realizado por investigadores de la Universidad de Bradenburgo [1] que hemos comentado antes se proporcionaban dos formas de reiniciar el contador.

- La primera forma consiste en **estar dos segundos sin enviar ningún comando**.
- La segunda forma consiste **en enviar un comando con el número de secuencia 1**, de esta manera reseteamos el contador.

Asimismo, en el trabajo de Mark Szabo descubrimos que al enviar el primer paquete, el dron **guarda la dirección IP del cliente** y solo aceptará paquetes que vengan de él. En el apartado ataques se describirá como podemos enviar paquetes suplantando la dirección IP del cliente y como todo este proceso es bastante sencillo.

Los desarrolladores del protocolo consideraron UDP como una mejor opción frente a TCP pues de esta manera, al no tener que esperar a que la conexión sea establecida, los paquetes se envían de forma más rápida, y si se pierde un paquete tampoco pasa nada, pues numerosas veces se envía una y otra vez el mismo paquete. En este caso, es más importante la velocidad que el asegurarse que un paquete llegue a su destino, y por ello es lógico que los desarrolladores optasen por esta opción. Pero al simplemente enviar el paquete, y no establecer una conexión como en TCP, este sistema resulta mucho más vulnerable frente a la suplantación de paquetes, y en definitiva, más inseguro.

3.4.2 Burlar el sistema de emparejamiento

Como bien hemos mencionado anteriormente, el dron nos da la opción (desactivada por defecto) de hacer el sistema de comunicaciones más seguro por medio del **emparejamiento**. El emparejamiento se activa desde el menú opciones, tal y como dijimos en la sección 3.2.2. Entonces el dispositivo envía el siguiente comando al dron:

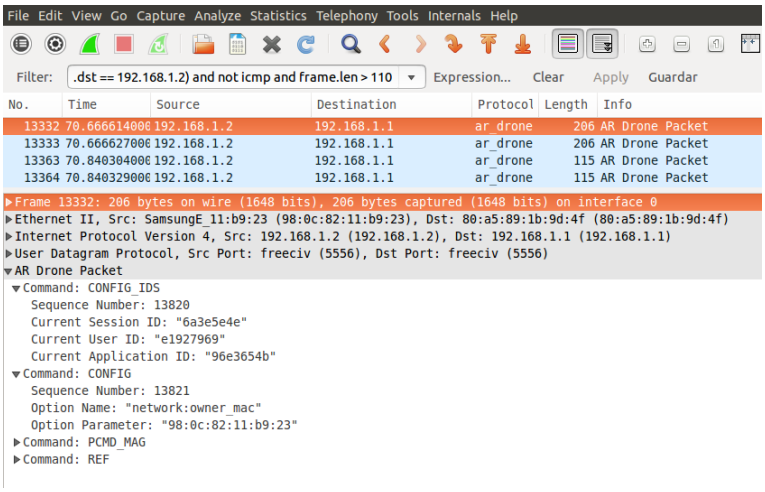


Figura 3.4.1: Orden de activar emparejamiento

A partir de ahora, todos los paquetes que **no tengan como dirección MAC origen la dirección MAC del dispositivo** que está emparejado que no sean **ICMP**, y que no vayan al puerto **21** o al **2049** (en realidad, aunque no se descarten los paquetes que van a este puerto, está cerrado, así que tampoco tiene mucha utilidad) serán descartados. El script que se encarga de establecer el emparejamiento es **/bin/pairing_setup.sh**. Este script básicamente comprueba que haya una dirección MAC con la que emparejar el dron distinta de 00:00:00:00:00:00, y si es así, haciendo uso del comando **iptables** establece que paquetes deben pasar y cuáles no.

Entonces la forma de suplantar un paquete con una orden de manejo tampoco se diferenciaría mucho de la manera en la que los suplantábamos cuando el dispositivo no estaba emparejado con el dron. En este caso, además de enviar un paquete con la dirección IP origen del cliente, habremos de hacerlo con la dirección MAC origen del cliente. También, utilizando este mecanismo de suplantación, podremos suplantar una orden que consista en desactivar el emparejamiento, y así poder acceder de nuevo a otros servicios, como Telnet.

En el apartado ataques se describirá detalladamente paso a paso como podemos conseguir burlar este sistema.

3.4.3 Telnet

Tras realizar el escáner de puertos con nmap, descubrimos que el dron tiene el puerto 23 (Telnet) abierto. Desde el terminal, iniciamos una sesión telnet y comprobamos que se abre una Shell como usuario root sin necesidad de introducir ninguna contraseña.

Una vez conectados al dron por medio de telnet, el número de posibles ataques a realizar es infinito, pues el único límite que existe es nuestra imaginación. En la sección ataques se comentarán algunos ataques que en una situación real a un atacante le podría resultar interesante realizar.

3.4.4 FTP

Realizando el escáner de puertos con nmap también descubrimos que el puerto 21 (FTP) está abierto. Nos conectamos desde la terminal y nos logueamos con el usuario **anonymous**.

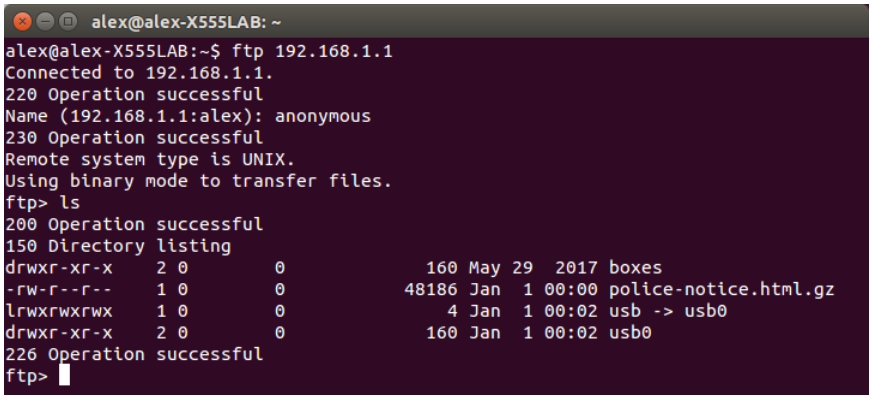


Figura 3.4.2: Conexión al dron a través de FTP

Como podemos ver, nos hemos logueado correctamente. Cuando el usuario conecta un dispositivo USB con el objetivo de guardar los vídeos y las fotos que ha hecho ahí, este dispositivo se monta en **usb0**. Nosotros entonces tendremos acceso al dispositivo USB y podremos realizar numerosos ataques, tales como introducir un archivo malicioso, o bien robar información confidencial.

3.4.5 Otras vulnerabilidades

En esta subsección se tratarán otras vulnerabilidades más complicadas de explotar y que por lo tanto no se les va a prestar la misma atención que a otras. Pero a pesar de ello, estas vulnerabilidades no dejan de existir y por lo tanto deben ser mencionadas en este trabajo.

3.4.5.1 Suplantación del flujo de vídeo del dron

El dron envía los paquetes que conforman el flujo de vídeo que el usuario ve en la pantalla de la aplicación a través del puerto 5555 a través del protocolo TCP. Si la opción de grabar está activada, envía el flujo de vídeo en H264-720p a través del puerto 5553 también sobre TCP. En ambos casos un atacante podría suplantar la señal de vídeo y conociendo el protocolo, enviar un vídeo que no tiene nada que ver con lo que ve la cámara del dron.

3.4.5.2 Instalar un firmware defectuoso e inutilizar el dron

Un atacante podría instalar un firmware defectuoso y dejar inutilizado el dron. Para ello deberá seguir los siguientes pasos:

1) Vamos a tener dos archivos de versión: **/firmware/versión.txt** (este archivo contiene la versión del firmware actual) y **/update/versión.txt** (este archivo contiene la versión del firmware que vamos a instalar). Modificamos ambos archivos de manera que el número de versión en /update/versión.txt sea superior al de /firmware/versión.txt.
2) Subimos el firmware que queremos instalar (en formato **.plf**) al dron, por medio del puerto **5551** (y no el 21) a través de FTP.
3) Desconectamos la batería durante un par de segundos y la volvemos a conectar. Comenzará la actualización. Tras unos minutos, habrá acabado y el nuevo firmware estará instalado.

Dado este procedimiento, a un atacante le resultaría extremadamente sencillo realizar este ataque.

3.5 Resumen

Vulnerabilidad	Descripción	Impacto
Comandos AT	El atacante puede enviar comandos para manejar el dron suplantando la IP del cliente.	Bajo, se soluciona activando el emparejamiento.
Comandos AT (emparejamiento activado)	Aunque activemos el *pairing*, el atacante sigue pudiendo enviar comandos para manejar el dron si suplanta la MAC del cliente. El emparejamiento puede desactivarse si se mandan los paquetes necesarios.	Alto, un atacante puede tomar el control del dron.
Telnet	Cualquiera puede conectarse e iniciar una sesión telnet como root sin necesidad de introducir contraseña.	Muy alto, con acceso a telnet un atacante domina el sistema.
FTP	Cualquiera puede conectarse por FTP sin necesidad de contraseña, y si un dispositivo USB está conectado, puede acceder a ese dispositivo.	Alto, si un dispositivo USB está conectado un atacante podría introducir, modificar o eliminar archivos.
Otras vulnerabilidades	Entre otras vulnerabilidades se encuentran la posibilidad de suplantar el flujo de vídeo del dron, y de instalar un fimware defectuoso para dejar el dron inutilizado.	Suplantar el flujo de vídeo del dron: medio. Instalar firmware defectuoso: muy alto.

Tabla 3.2: Resumen de las vulnerabilidades encontradas

4 Desarrollo. Ataques

Tras haber realizado un análisis de vulnerabilidades en el apartado anterior, en este apartado se desarrollarán distintos ataques aprovechando las distintas vulnerabilidades encontradas.

4.1 Suplantación de órdenes de manejo del dron. Parte 1

En el apartado 3.4.1 explicamos desde un punto de vista teórico como podíamos suplantar paquetes con comandos AT y de esta manera enviar órdenes de movimiento al dron. En esta subsección realizaremos este ataque siguiendo los planteamientos antes enunciados.

Desde nuestro dispositivo móvil (o tablet) nos conectamos a la red Wi-Fi del dron, abrimos la aplicación AR.FreeFlight y vamos a la opción Take Off y desde ahí comenzamos a volar el dron.

Se ha desarrollado el siguiente script **at_spoof.py** [26] en Python haciendo uso de la librería **Scapy**, el cual enviará paquetes con órdenes al dron para que aterrice mientras intentamos manejarlo con el teléfono móvil.

Este script envía paquetes (hasta alcanzar los 10000) con el código de aterrizaje. El número de secuencia siempre será 1, pues de esta manera reseteamos el contador y no nos tenemos que preocupar porque los paquetes tengan un número de secuencia en orden ascendente. El script se ejecuta con el siguiente formato:

sudo python at_spoof.py ip_dispositivo

La IP dispositivo será la dirección IP origen de los paquetes, pues como ya dijimos, el dron solo tendrá en cuenta los paquetes que tengan la IP del primer dispositivo que haya comenzado a enviar comandos.

Ahora nos conectamos desde el portátil a la red Wi-Fi del dron. Por medio de nmap (aplicando la técnica que hemos visto en el apartado 3.3.3) averiguaremos la dirección IP del dispositivo, que en este caso se corresponde con **192.168.1.2**. Ahora, mientras el dron está volando, ejecutamos el script.

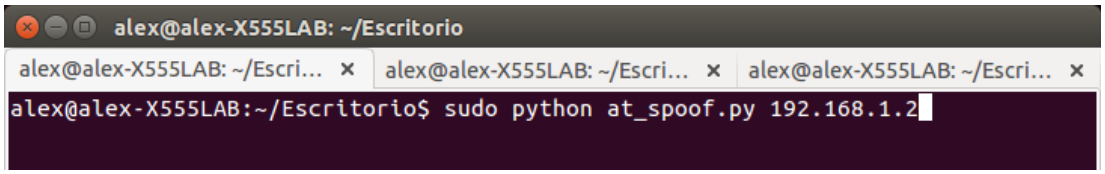


Figura 4.1.1: Ejecución del script at_spoof.py

Y podremos observar como el dron, el cual estaba en movimiento, aterriza. El proceso se puede ver en el siguiente vídeo [25].

4.2 Suplantación de órdenes de manejo del dron. Parte 2. Cómo burlar el sistema de emparejamiento

Tal y como dijimos en la sección vulnerabilidades, el dron nos da una opción para poder hacer el sistema (algo) más seguro: el **emparejamiento**. Cuando activamos la opción de emparejar, se envía el comando (tal y como vimos en la sección vulnerabilidades):

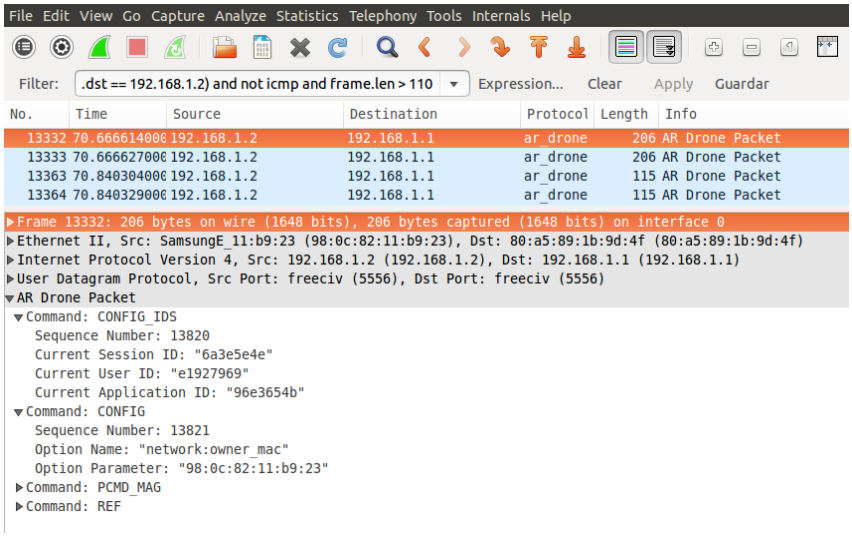


Figura 4.2.1: Comando para activar el emparejamiento

A partir de ahora, todos los paquetes (salvo los paquetes ICMP, y los que vayan al puerto 21 y 2049) que no tengan como dirección MAC origen la dirección MAC del dispositivo serán descartados. Si probamos volar el dron, y ejecutar el script del apartado anterior, veremos que no surte ningún efecto, puesto que los paquetes son descartados al tener como MAC origen la MAC del atacante.

Sin embargo, tampoco en este caso se libraría el cliente de ningún ataque. Para suplantar órdenes de manejo del dron bastaría con suplantar la MAC origen del paquete por la MAC del dispositivo que está manejando el dron.

Comencemos entonces con nuestro ataque. Si estábamos conectados a la red Wi-Fi antes de activar la opción de emparejamiento, ya tendremos una dirección IP asignada. Pero si no lo estábamos tenemos un pequeño problema: **el puerto 67**, donde se ejecuta el servicio DHCP y desde donde se asignan direcciones IP dinámicas, **bloqueará nuestras peticiones**, tal y como hacen la mayoría de los puertos. La solución aquí también será trivial: tras conectarnos a la red Wi-Fi, configurando una dirección IP estática esto dejará de ser un obstáculo (**esto solo hará falta hacerlo si nos hemos conectado a la red Wi-Fi después de activarse la opción de emparejamiento**). En el **Anexo C** puede encontrarse más información de cómo configurar una dirección IP estática para esta conexión.

El siguiente paso será averiguar la dirección IP del dispositivo, la dirección MAC del dispositivo, y la dirección MAC del dron. Para ello, una vez más volveremos a utilizar nuestra preciada herramienta nmap. Ejecutamos el siguiente comando (es importante el sudo pues sino en los resultados de nmap no se mostrará la MAC):

sudo nmap –sn 192.168.1.0/24

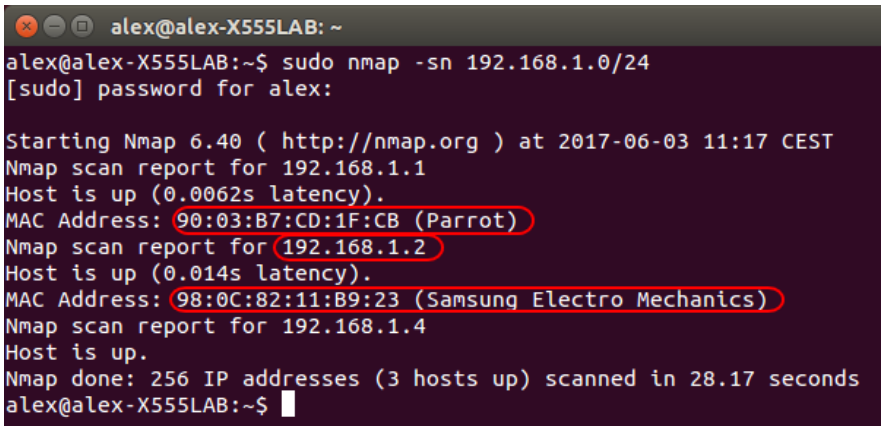


Figura 4.2.2: Escaneo de hosts para averiguar las direcciones MAC

Como podemos ver, la dirección MAC del dron es **90:03:B7:CD:1F:CB**, la dirección MAC del dispositivo es **98:0C:82:11:B9:23**, y la dirección IP del dispositivo es **192.168.1.2** (la IP del dron siempre va a ser 192.168.1.1 como hemos dicho repetidas veces por lo que no nos debemos preocupar por ello).

Una vez hecho esto, comenzamos a volar el dron. Cuando el dron ya esté volando, tratamos de ejecutar el script **at_spoof.py** [26].

sudo python at_spoof.py ip_dispositivo

Vemos que no da ningún resultado, pues al ser la dirección MAC origen distinta a la dirección MAC del dispositivo, los paquetes serán descartados. Con el fin de burlar esta barrera de seguridad, se ha desarrollado un nuevo script **at_eth_spoof.py** [27] el cual además de suplantar la dirección IP del dispositivo, suplanta su MAC.

El formato de ejecución del script es el siguiente:

sudo python at_eth_spoof.py ip_dispositivo MAC_dispositivo MAC_dron

Ejecutamos el script con los parámetros que hemos averiguado anteriormente. Y como podemos observar, esta vez el dron si aterriza.

Ahora bien, si bien por medio de esta técnica podemos suplantar las órdenes al dron, así como cualquier otro paquete UDP, no podremos acceder a aquellos servicios que corran sobre TCP, tales como Telnet. Pero de la misma manera que mediante un comando AT se activa el emparejamiento, enviando ese mismo comando y la dirección MAC **00:00:00:00:00:00** se podrá desactivar, y así tendremos carta blanca para hacer lo que queramos con el dron. Pero deberemos conocer las IDs que se envían en los comandos de configuración (en rojo) y enviar los IDs de configuración junto con el comando.

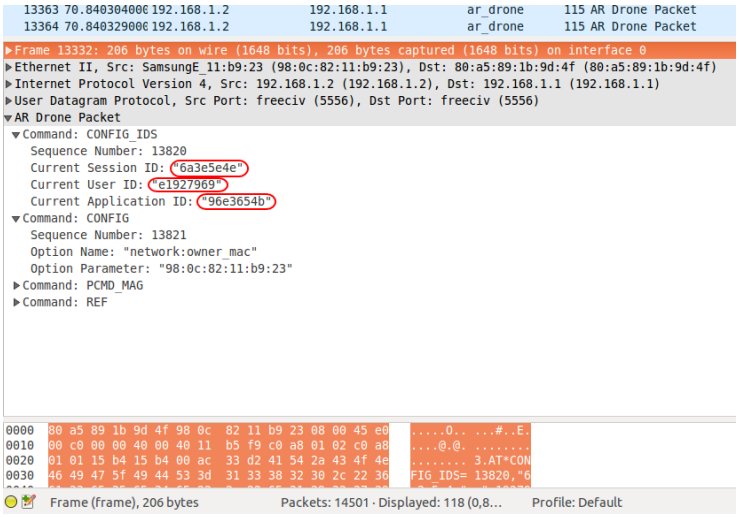


Figura 4.2.3: Claves para activar el emparejamiento

Como solución a este problema se ha desarrollado un script el cual el mismo se encarga de realizar un ataque de envenenamiento ARP, y de esnifar los paquetes hasta conseguir las IDs que se envían en los comandos de configuración. Una vez se tengan esas IDs, el programa envía varios (para asegurarnos) paquetes con la orden de desactivar el emparejamiento. Hecho esto, el emparejamiento ha quedado desactivado y el dron es nuestro. Ya podremos acceder a telnet o a otros servicios a los que antes no podíamos.

El script se encuentra aquí [28] y su formato de ejecución es el siguiente:

sudo python deactivate_mac_filter.py ip_dispositivo MAC_dispositivo MAC_dron

En este vídeo [29] se puede ver como se burla este sistema de seguridad de manera práctica.

4.3 Ataques al dispositivo USB conectado al dron. Explotando el FTP

Como bien indicamos en el apartado 3.4.4, cuando un usuario conecta un dispositivo USB con el objetivo de guardar los vídeos y las fotos que se hagan con el dron ahí, ese dispositivo se monta en **usb0**. Y si nos conectamos a través de FTP, podremos acceder a ese dispositivo:

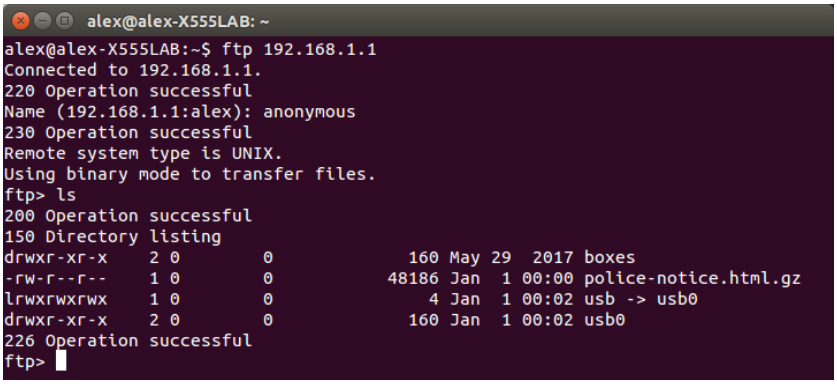


Figura 4.3.1: Acceso a FTP desde el dron

Teniendo acceso al dispositivo, las posibilidades a la hora de realizar ataques son infinitas. Entre estos ataques se encontrarían: **robar uno o más archivos confidenciales**, **modificar algún archivo de suma importancia**, e **introducir un archivo malicioso**. En el **Anexo D** se ponen ejemplos prácticos de estos ataques.

4.4 Ataques a través del servicio telnet

Como bien dijimos en el apartado 3.4.3, el dron nos da la posibilidad de conectarnos a él a través de telnet y podremos iniciar una sesión como root sin necesidad de introducir usuario o contraseña. Una vez conectados, el número de ataques que se pueden realizar es infinito. En este apartado se describirán algunos ataques los cuales un atacante estaría especialmente interesado en realizar, aunque por supuesto puede haber muchos más.

4.4.1 Infectar automáticamente cualquier dispositivo USB que se conecte.

En el apartado 4.3.3 vimos como a través de FTP podíamos introducir un archivo malicioso a un dispositivo USB que se conectase al dron. En este apartado veremos cómo modificando algunos archivos del sistema se puede conseguir que cualquier dispositivo USB que se conecte sea automáticamente infectado.

El script **/etc/init.d/rcS** se ejecuta cada vez que se enciende el dron. Se ha desarrollado un script **infection.sh** [32] el cual se ejecuta con ./infection.sh start y el cual se para con ./infection.sh stop. Este script consiste en un bucle infinito el cual comprueba si existe la carpeta /data/video/usb (si existe, es que hay montado un dispositivo USB) y copia el troyano (que debe estar en la carpeta /bin) al dispositivo USB, imitando el formato que tienen los vídeos. El script espera un intervalo de 10 segundos entre cada iteración del bucle.

Lo primero que deberíamos hacer es conectarnos a través de FTP al dron, y subir al directorio raíz el archivo troyano.exe del que hemos hablado anteriormente.

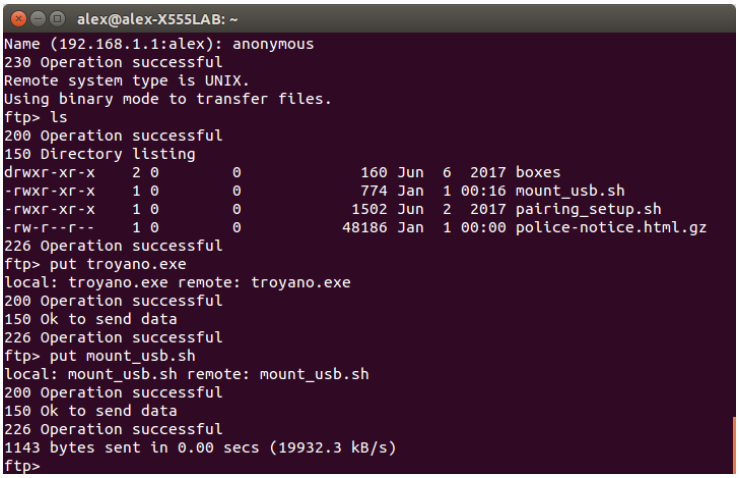


Figura 4.4.1: Subida de un archivo malicioso a través de FTP

El directorio raíz cuando nos conectamos por FTP se corresponde con el directorio /data/video/ del dron. Una vez hemos subido nuestro archivo malicioso al dron, copiamos el

archivo de la carpeta /data/video a la carpeta /bin (aunque podría ser cualquier otra carpeta siempre y cuando modifiquemos el script).

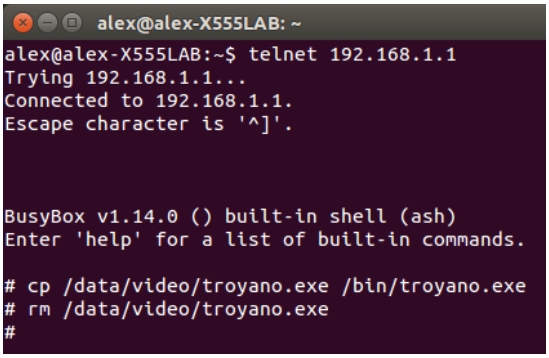


Figura 4.4.2: Copia del archivo malicioso a la carpeta correspondiente

Ahora subimos nuestro script infection.sh y lo movemos a /etc/init.d/:

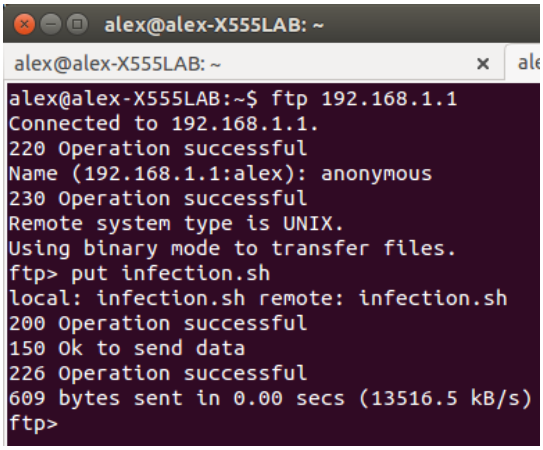


Figura 4.4.3: Subida del archivo infection.sh

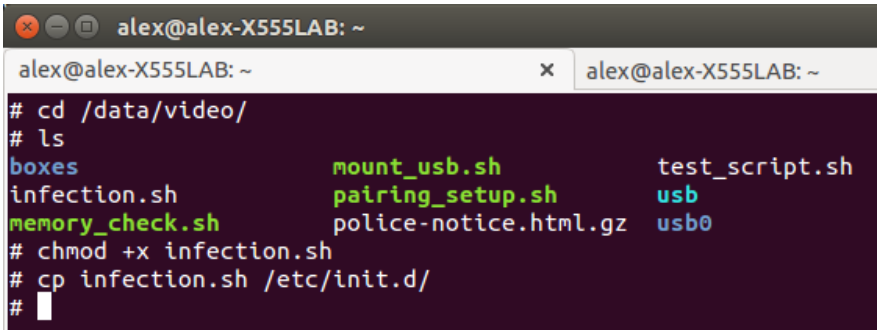


Figura 4.4.4: Movimiento del archivo infection.sh a /bin

Ahora abrimos el archivo /etc/init.d/rcS y editamos la penúltima línea (antes de echo end init…), de manera que quede así:

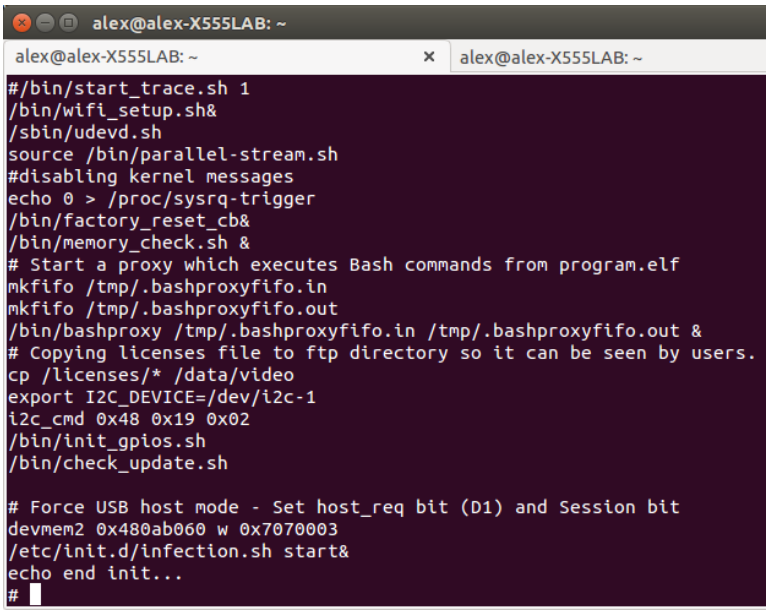


Figura 4.4.5: Edición del archivo /etc/init.d/rcS

Guardamos, y la próxima vez que reiniciemos el dron, nuestro script se ejecutará en segundo plano. Una vez hemos reiniciado, conectamos un dispositivo USB al dron, y vamos a la carpeta /data/video/usb/. Ejecutamos ls y vemos que se han creado nuevas carpetas que antes no existían y que responden al formato de carpetas que crea el script. Esperamos 10 segundos y volvemos a ejecutar ls. El resultado es el siguiente:

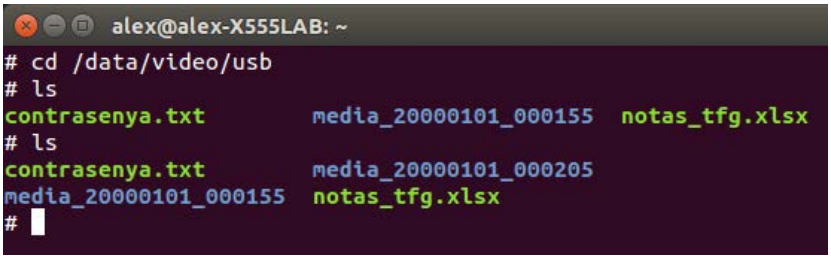


Figura 4.4.6: Infección de un dispositivo USB

Ahora, cada vez que un dispositivo USB se conecte al dron, será infectado. En caso de que se desee parar la ejecución del script, bastaría con ejecutar **/etc/init.d/infection.sh stop**.

4.4.1 Desactivar el *pairing* definitivamente

Cuando el dispositivo envía al dron el comando para activar el emparejamiento, el dron ejecuta el script **/bin/pairing_setup.sh**. Podemos modificar el script de tal manera que a partir de ahora cuando se active el emparejamiento, no se bloqueará el acceso a ningún puerto. Para ello basta modificar la línea 46 (haciendo uso de vi) y sustituir DROP por ACCEPT:

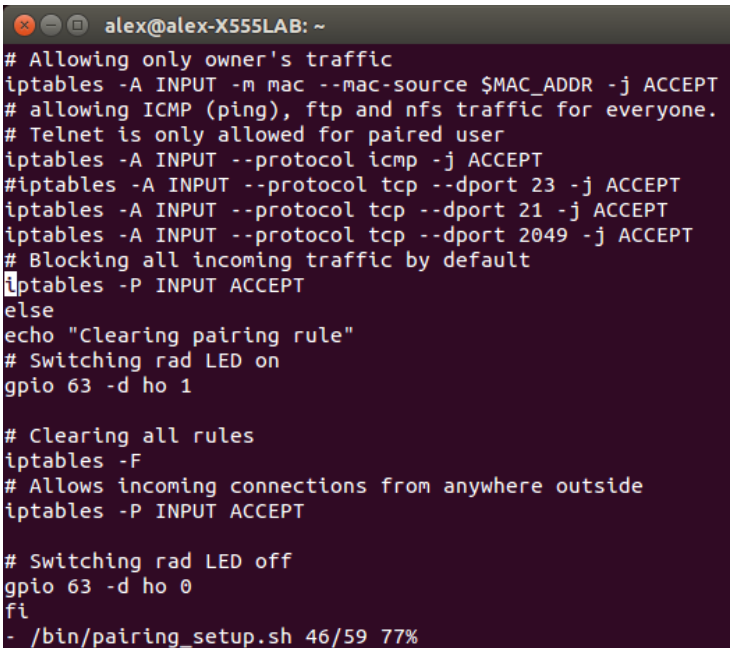


Figura 4.4.7: Edición del archivo pairing_setup.sh

Una vez hecho esto, cada vez que cualquier dispositivo intente emparejarse con el dron, el emparejamiento no servirá de nada.

4.4.2 Modificar la configuración del dron

En el archivo **/data/config.ini** se encuentra la configuración del dron. Cualquier atacante podría entrar a través de telnet y modificar este archivo, estableciendo las opciones de su preferencia.

4.4.3 Otros ataques. Ataques destructivos.

Los ataques que se han tratado en este apartado solo son algunos ataques que un atacante podría estar interesado en realizar, aunque por supuesto puede haber muchos más. Entre estos ataques podrían encontrarse ataques destructivos, que consistirían simplemente en borrar y modificar archivos del sistema con el fin de dañarlo. Este tipo de ataques no se han tratado aquí puesto que se ha preferido centrarse en ataques de los que el atacante puede obtener algo útil, y en una situación real un atacante no estaría interesado en realizar este tipo de ataques. Sin embargo, no por ello estos ataques dejan de ser posibles.

4.5 Resumen

Vulnerabilidad	Ataque
Comandos AT	Ayudándonos de un script en Python, enviamos órdenes al dron para que aterrice suplantando la dirección IP. [26]
Comandos AT (emparejamiento activado)	Ayudándonos de un script en Python, enviamos órdenes al dron para que aterrice suplantando la dirección IP y la MAC. [27]
	Se ha desarrollado un script en Python que esnifa paquetes enviados desde el cliente al dron hasta encontrar aquellos con las claves necesarias para desactivar el emparejamiento. Entonces se envía la orden de desactivar el emparejamiento. [28]
FTP	Robar archivos confidenciales.
	Modificar archivos de suma importancia.
	Introducir un archivo malicioso.
Telnet	Infectar automáticamente cualquier dispositivo USB que se conecte. [32]
	Desactivar el *pairing* definitivamente.
	Modificar la configuración del dron.
	Ataques destructivos.

Tabla 4.1: Resumen de ataques

5 Contramedidas

En esta sección se propondrán y se explicarán detalladamente una serie de contramedidas para solucionar los distintos fallos de seguridad descubiertos en el dron.

5.1 Conexión Wi-Fi segura

En el trabajo realizado por los investigadores de la Universidad de Bradenburgo [1], se propone un método para hacer la conexión entre el dron y el dispositivo más segura. El método consiste en **generar una red Wi-Fi segura** (**WPA** o **WPA2**) desde el terminal y que sea el dron quien se conecte a esa red, en vez de conectarse el terminal a la red Wi-Fi del dron.

Para que el dron sea capaz de conectarse a una red Wi-Fi WPA o WPA2 debemos instalar la herramienta **WPA Supplicant**, la cual permitiría al dron conectarse a una red Wi-Fi con este tipo de seguridad. Sin embargo, el dron **tiene otra arquitectura**, concretamente ARM, así que deberemos descargarnos el código fuente, modificar algunos parámetros de configuración a la hora de compilar, y finalmente **compilarlo de manera cruzada**. Una vez compilado, moveríamos los ejecutables y todos los recursos necesarios a la carpeta /bin del dron. En el trabajo realizado por estos investigadores se describe el proceso paso a paso.

Una vez instalada esta herramienta en el dron, el dron podrá conectarse a una red Wi-Fi segura. El proceso de conexión se haría de la siguiente manera:

1) El terminal desde el cual se va a manejar el dron genera una red Wi-Fi segura.
2) Un tercer dispositivo se conecta a la red Wi-Fi abierta del dron e inicia una sesión telnet.
3) El tercer dispositivo envía una serie de comandos al dron para que este se conecte a la red Wi-Fi segura, indicando el ESSID, la contraseña y estableciendo la IP del dron como 192.168.1.1. Esto último es importante, pues de otra manera no se va a poder manejar el dron desde la aplicación oficial.
4) Una vez el dron esté conectado a la red Wi-Fi segura, podremos manejar el dron desde la aplicación oficial tal y como hacíamos antes.

Este método a priori parece que hace la conexión entre el dron y el terminal algo más segura, y además presenta una serie de ventajas adicionales, puesto que la nueva red Wi-Fi dispondrá de conexión a Internet y ahí se nos abre un mundo lleno de posibilidades. Sin embargo también presenta importantes desventajas.

En primer lugar, el proceso que debemos realizar para que el terminal se conecte a una red Wi-Fi segura es bastante tedioso. Es cierto que se podría automatizar por medio de algún script, pero en cualquier caso haría falta un tercer dispositivo para llevar a cabo este proceso, y no siempre vamos a disponer de él.

En segundo lugar, para enviar comandos al dron con la orden de que éste se conecte a una red Wi-Fi segura deberemos, antes de nada, conectarnos a la red Wi-Fi insegura del dron. Algunos ataques podrían realizarse justo en ese momento. Sin embargo, el problema es aún

más grave: en los comandos que enviamos al dron para que éste se conecte a la red Wi-Fi segura tenemos que indicar si o si la contraseña de dicha red Wi-Fi. Un atacante podría esnifar estos paquetes haciendo uso de Wireshark (o de otra herramienta similar) y de esta manera averiguar la contraseña de la red Wi-Fi. Y con la clave de esta red Wi-Fi en su poder, un atacante podrá llevar a cabo los mismos ataques que llevaría a cabo en una red Wi-Fi abierta.

Una solución mejor consistiría en, en vez de hacer que el dron se conecte a una red Wi-Fi segura generada por el terminal, hacer que el dron genere su propia red Wi-Fi segura. Pero al menos con el firmware dado, el dron no nos permite esa opción. Para poder generar una red Wi-Fi segura deberíamos actualizar o modificar el firmware del dron, y si no se tiene experiencia en el asunto, se corre el riesgo de dejar el dron inutilizado.

5.2 Filtrado de MAC

Una manera de incrementar el nivel de seguridad de la conexión entre el dron y el terminal consistiría en, cuando se activa el emparejamiento, no solo bloquear los paquetes que entren a ciertos puertos y no tengan la dirección MAC del dispositivo emparejado, sino establecer el **filtrado de MAC a nivel de Wi-Fi**. En este caso un dispositivo que no tenga la dirección MAC del dispositivo emparejado, directamente ni podrá conectarse a la red Wi-Fi del dron.

El gran problema a la hora de llevar esto a cabo es el mismo que el de establecer WPA/WPA2 como sistema de seguridad para la red Wi-Fi: el sistema operativo del dron no soporta esta opción, y habría que realizar serias modificaciones en el firmware.

Aun estableciéndose esta medida de seguridad, el atacante podría **clonar la MAC** del cliente y conectarse a la red Wi-Fi. En este caso el dron detectaría que hay dos dispositivos con la misma MAC y se produciría un error de conexión. El cliente será incapaz de interactuar con el dron. Aunque la comunicación entre el dron y el terminal aún puede sufrir este ataque, al menos el atacante no podrá realizar ataques que pudieran tener consecuencias más graves.

5.3 Contraseña del root y SSH

Como bien hemos indicado en repetidas ocasiones a lo largo de este trabajo, el puerto 23 está abierto y el puerto telnet está escuchando en él. Si estamos conectados a la red Wi-Fi del dron, podemos iniciar una sesión telnet en la que accederemos como **root** sin ni siquiera tener que introducir una contraseña.

Para establecer/cambiar la contraseña del usuario root en Ubuntu, utilizamos el comando **passwd**. Sin embargo, al teclear este comando podremos ver como el sistema nos informa de que ese comando no existe. En el sistema operativo del dron, no existe ninguna herramienta para poder establecer/cambiar la contraseña del root.

No nos queda otra que editar manualmente el archivo **/etc/passwd** e introducir ahí la contraseña del root que deseemos encriptada en SHA1. Sin embargo, no es recomendable tratar de modificar el fichero /etc/passwd si no se conoce bien cómo funciona el sistema y como le puede afectar el modificar este fichero.

El hecho de haber establecido una contraseña para el usuario root no es suficiente para que la comunicación a través de telnet sea segura. Cuando un usuario se autentica, el atacante

podría obtener los datos de autenticación haciendo uso de un *sniffer*. Una solución frente a este problema consistiría en usar **SSH** en vez de telnet.

Para ello nos descargamos el código fuente de algún servidor SSH (como OpenSSH o Dropbear) y lo compilamos de manera cruzada, pues la arquitectura del dron es ARM. En este enlace tenemos una guía de cómo compilar un programa de manera cruzada para ARM [30].

Una vez compilado el servidor SSH, podremos subirlo al dron por medio de FTP y moverlo a la carpeta /bin (donde se encuentran todos los ejecutables) desde una sesión telnet. Asimismo podremos hacer que el servidor SSH se ejecute en segundo plano cada vez que se encienda el dron modificando el archivo /etc/init.d/rcS. Y podremos bloquear el acceso a telnet editando el fichero /bin/pairing_setup.sh.

Sin embargo, es muy probable que al instalar un servidor SSH en el dron uno tenga problemas puesto que un servidor SSH requiere de ciertas librerías que en la versión compacta de Ubuntu que tiene el dron posiblemente no se encuentren. Una vez más, nos topamos con el problema del firmware.

5.4 SFTP

En apartados anteriores vimos como podíamos conectarnos al servidor FTP de manera completamente anónima (no necesitábamos usuario ni contraseña) y si un dispositivo USB estaba conectado al dron, teníamos acceso completo a él. Dada esta vulnerabilidad un atacante podría realizar numerosos ataques, que ya se mencionaron anteriormente.
Si para solucionar el problema de telnet instalábamos SSH, para solucionar este problema instalaremos **SFTP**.

Al igual que en SSH, uno puede encontrar numerosos servidores SFTP en Internet. Ya que el proceso de instalación de un programa en el dron es algo tedioso (al tener otra arquitectura y al ser el SO una versión compacta), una buena opción sería instalar OpenSSH, y una vez instalado, simplemente habría que cambiar algunas opciones de configuración para poder usar SFTP. En este tutorial se indica paso a paso como hacerlo [31].

Ahora bien, las limitaciones del firmware siguen existiendo y en esta ocasión, no han dejado de ser un problema.

5.5 Resumen de contramedidas

En la siguiente tabla se muestra de manera resumida el servicio vulnerable, que fallos de seguridad tiene, y la solución propuesta.

Servicio	Problema	Solución
Red Wi-Fi	Red Wi-Fi abierta, lo cual de por sí facilita que se puedan realizar otros ataques.	Establecer en la red Wi-Fi un sistema de seguridad WPA/WPA2.
Envío de órdenes al dron	Aunque se active el *pairing*, se puede suplantar la dirección MAC para enviar órdenes al dron, e incluso para desactivar el *pairing*.	Filtrado de MAC. No permitir que nadie, nada más que el dispositivo con la MAC emparejada, se conecte.
Telnet	Si iniciamos una sesión telnet accedemos como root sin necesidad de introducir ninguna contraseña.	Establecer una contraseña para el root. Instalar SSH.
FTP	Acceso anónimo (sin usuario ni contraseña), con acceso completo a cualquier dispositivo USB que esté conectado.	Instalar SFTP.

Tabla 5.1: Resumen de contramedidas

6 Conclusiones y trabajo futuro

6.1 Conclusiones

En este trabajo se ha estudiado en detalle la seguridad que existe en las comunicaciones entre un dron y un cliente, tomando como muestra representativa el dron AR Drone 2.0.

En primer lugar, una vez comprendido el funcionamiento básico del dron, se ha realizado un **análisis de vulnerabilidades** del dron y de la comunicación entre el dron y el cliente, utilizando las herramientas típicas que se utilizan en este tipo de análisis, y realizando todos los pasos que se suelen seguir, tales como escáner de puertos, envenenamiento por ARP y analizar los paquetes que se envían de un lado a otro ayudándonos de un *sniffer*, etc. Una vez descubiertas las vulnerabilidades, se han descrito una por una.

En segundo lugar, se han **descrito detalladamente los distintos ataques** que se pueden realizar explotando las vulnerabilidades descubiertas anteriormente, pero siempre centrándonos en aquellos ataques que en una situación real un atacante puede estar interesado en explotar, y no aquellos que aun explotando las vulnerabilidades descubiertas, no se obtiene provecho alguno. Asimismo, se ha insertado enlaces donde se muestran vídeos de cómo realizar estos ataques, y también se ha insertado enlaces a los diversos scripts que explotan estas vulnerabilidades.

En último lugar, se han **propuesto y explicado una serie de contramedidas** para tratar de tapar aquellos agujeros de seguridad que se han encontrado en la comunicación entre el dron y el cliente.

La conclusión final de este trabajo no deja de ser una confirmación de la hipótesis inicial: **mientras en los últimos años en el área de los drones se ha avanzado a un ritmo bastante alto, en el campo de la seguridad no se ha avanzado al mismo ritmo**. Este trabajo debe servir para, entre otras más cosas, no solo describir los diversos fallos de seguridad que existen en las comunicaciones de los drones, sino **concienciar** de que la seguridad no se puede dejar de lado y se debe desarrollar de manera paralela al desarrollo general de los drones.

6.2 Trabajo futuro

Respecto a trabajos futuros, sería interesante tener en cuenta los siguientes puntos:

- Los **ataques de suplantación de la señal de vídeo y de actualización de firmware** con el fin de dejar al dron inoperativo se han descrito brevemente, pero no se han desarrollado. Sería interesante estudiar estos ataques más en profundidad y describir cómo se llevarían a cabo.
- En este trabajo se han propuesto una serie de contramedidas para tapar los agujeros de seguridad. Un trabajo futuro podría consistir en **implementar estas contramedidas y describir paso a paso como se implementarían**.
- El dron que se ha escogido para llevar a cabo este trabajo es una muestra de los drones que hay actualmente en el mercado, pero sería interesante no utilizar una

única muestra, sino varias. Un trabajo futuro podría consistir en, siguiendo los mismos pasos que en este trabajo, **estudiar la seguridad en los drones más representativos del mercado**.

- A la hora de suplantar órdenes de manejo del dron, se ha estudiado en profundidad como realizar este proceso, sin embargo, los scripts que se han desarrollado para explotar esta vulnerabilidad son bastante rudimentarios. Sería interesante **implementar una aplicación, que, suplantando órdenes de un cliente, permita manejar el dron tal y como se maneja desde la aplicación oficial**.

Referencias

[1] Hacking and securing the AR.Drone 2.0 quadcopter - Investigations for improving the security of a toy
https://www.researchgate.net/publication/260420467_Hacking_and_securing_the_ARDrone_20_quadcopter_-_Investigations_for_improving_the_security_of_a_toy
[2]
Research presented at the Ethical Hacking Conference Budapest in May 2016 – Mark Szabo
https://github.com/markszabo/drone-hacking
[3] Samy https://en.wikipedia.org/wiki/Samy_(computer_worm)
[4] Skyjack: video https://www.youtube.com/watch?v=EHKV01YQX_w
[5] Skyjack: más información http://samy.pl/skyjack/
[6] Características de Parrot Bebop 2 FPV https://www.parrot.com/es/drones/parrot-bebop-2-fpV#técnicos
[7] Parrot Bebop 2 – Tutorial #1 – Set-up
https://www.youtube.com/watch?v=txL3imXuYSA
[8] DJI Phantom 3 https://www.dji.com/es/phantom-3-pro
[9] Drones más populares
https://www.amazon.es/s/ref=sr_ex_p_36_0?rh=n%3A599385031%2Ck%3Adrones&sort=popularity-rank&keywords=drones&ie=UTF8&qid=1495486366
[10] Como funciona el controlador remoto del DJI Phantom 3
http://wiki.dji.com/en/index.php/Phantom_3_Professional-Remote_Controller
[11] 3DR Solo https://3dr.com/solo-drone/specs/
[12] NincoAir Stratus WiFi GPS
http://www.ninco.com/es/8960-nincoair-quadrone-stratus-wifi-gps.html
[13] Parrot Bebop https://www.xataka.com/drones/parrot-bebop-drone-analisis
[14] Hubsan H501S
https://www.amazon.es/H501S-BRUSHLESS-Cuadricoptero-Headless-Auto-retorno/dp/B019N666K4/ref=sr_1_4?s=toys&ie=UTF8&qid=1495492629&sr=1-4&keywords=drones
[15] Hubsan H501S – Vídeo https://www.youtube.com/watch?v=sGIom61bdM0
[16] Syma X5C http://www.symatoys.com/goodshow/x5c-syma-x5c-explorers.html
[17] GoolRC T5G http://www.goolrc.com/rc-quadcopter-1192/p-rm5094r.html
[18] OCDAY M65500
https://www.amazon.es/OCDAY-Juguetes-educativos-M65500-M700-Negro/dp/B06XPXSYVG/ref=sr_1_4?s=toys&ie=UTF8&qid=1495547061&sr=1-4&keywords=drones
[19] KYG JJRC H36 https://www.amazon.es/KYG-Girocomp%C3%A1s-Antiaplastamiento-Interruptor-Velocidad/dp/B01KTEWIG0/ref=sr_1_2?s=toys&ie=UTF8&qid=1495548746&sr=1-2&keywords=drones
[20] EACHINE E010 Mini UFO https://www.amazon.es/EACHINE-Cuadric%C3%B3ptero-Headless-Teledirigido-Quadcopter/dp/B01KHV5O1G/ref=sr_1_4?s=toys&ie=UTF8&qid=1495548746&sr=1-4&keywords=drones
[21] RC Quadcopter

https://www.amazon.es/Quadcopter-funci%C3%B3n-retenci%C3%B3n-FPVRC-helic%C3%B3ptero/dp/B01N2K3BO3/ref=sr_1_7?s=toys&ie=UTF8&qid=1495548746&sr=1-7&keywords=drones
[22] AR Drone Developer Guide https://jpchanson.github.io/ARdrone/ParrotDevGuide.pdf
[23] Características técnicas del AR Drone https://www.parrot.com/es/drones/parrot-ardrone-20-elite-edition#técnicos
[24] AR.Freeflight
https://play.google.com/store/apps/details?id=com.parrot.freeflight&hl=es
[25] Ataques al dron AR Drone 2.0. Suplantación de órdenes de manejo del dron. https://www.youtube.com/watch?v=ElNovPg44T8
[26] Script at_spoof.py: suplantación de órdenes de manejo del dron https://github.com/alexero6/ardronehacking/blob/master/at_spoof.py
[27] Script at_eth_spoof.py: suplantación de órdenes de manejo del dron. MAC spoofing.
https://github.com/alexero6/ardronehacking/blob/master/at_eth_spoof.py
[28] Script deactivate_mac_filter.py: Esnifa paquetes hasta averiguar los ids de configuración y desactiva el *pairing*
https://github.com/alexero6/ardronehacking/blob/master/deactivate_mac_filter.py
[29] Ataques al dron AR Drone 2.0. Como burlar el *pairing*.
https://www.youtube.com/watch?v=kJtCdq5RDz4
[30] Compilación cruzada https://www.acmesystems.it/arm9_toolchain
[31] SFTP http://usandolarueda.blogspot.com.es/2013/01/crear-un-acceso-sftp-en-ubuntu.html
[32] Infection.sh https://github.com/alexero6/ardronehacking/blob/master/infection.sh

Glosario

Pairing	Emparejamiento. Asociación entre el dron y el cliente.
Sniffer	Software capaz de leer todos los paquetes que pasan por un host.
Escáner de puertos	Software que analiza un host local o remoto con el fin de encontrar puertos abiertos.
Comandos AT	Comandos en formato texto que son enviados al dron con el fin de controlarlo.

Anexos

A Especificaciones técnicas del dron

A.1 Asistencia electrónica

- Procesador de 32 bits con una frecuencia de 1 GHz basado en ARM Cortex A8.
- GPU PowerVR SGX530 con una frecuencia de 800 MHz.
- Sistema operativo Linux 2.6.32
- Memoria RAM DDR2 1 GB a 200 MHz
- USB 2.0 de alta velocidad para las extensiones
- Wi-Fi b g n
- Giroscopio: 3 ejes, precisión de 2000°/segundo
- Acelerómetro: 3 ejes, precisión de ± 50 mg
- Magnetómetro: 3 ejes, precisión de 6°
- Sensor de presión: Precisión de ± 10 Pa
- Sensores de ultrasonidos para medir la altitud: Medición de la altitud
- Cámara vertical: QVGA 60 FPS para medir la velocidad en vuelo

A.2 Motorización

- 4 motores sin escobillas de tipo "inrunner": 14,5 vatios y 28 500 rpm
- Rodamiento de bolas en miniatura: Sí
- Engranajes Nylatron: Sí
- Rodamiento de bolas autolubricante de bronce: Sí

A.3 Cámara

- Cámara HD 720p 30 FPS
- Objetivo gran angular: diagonal 92°
- Perfil de codificación básica: H264
- Formato fotos: JPEG
- Conexión: Wi-Fi

A.4 Peso

- Con carena interior: 380 g
- Con carena exterior: 420 g

B Manejo del dron

El dron se maneja desde una aplicación llamada **AR.FreeFlight**, disponible en Google Play [24]. Lo primero que debemos hacer para manejar el dron (después de descargarnos la aplicación) es conectarnos a la red Wi-Fi del dron, cuyo ESSID es **ardrone2**.

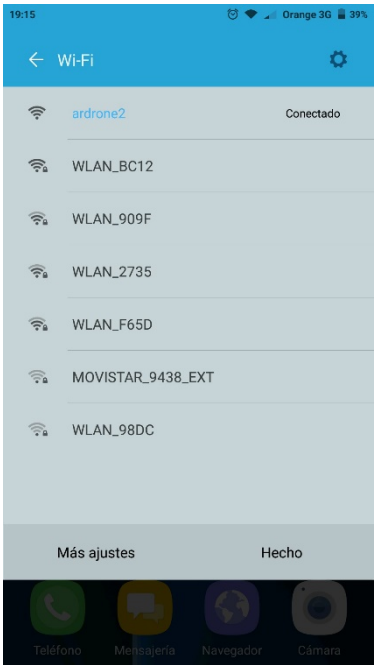


Figura B.1: Conexión a la red Wi-Fi ardrone2 desde el móvil

Una vez nos hemos conectado, abrimos la aplicación, y vemos el siguiente menú:

Figura B.2: Inicio de aplicación AR FreeFlight

Nos aparecen las siguientes opciones:

- **Take off**. Desde aquí es desde donde manejaremos al dron.
- **AR.Drone Academia**. Es una especie de red social donde puedes compartir tus logros y estadísticas volando, así como vídeos y fotografías.
- **Fotos/vídeos**. Aquí se guardan nuestras fotos y nuestros vídeos.
- **Actualización AR.Drone**. Desde aquí se actualizaría el firmware del dron.
- **Vídeos de usuarios**. Aquí puedes ver algunos vídeos que han subido otros usuarios.

Para acceder a las opciones AR.Drone Academia y Vídeos de usuarios debes disponer de conexión a Internet, por lo tanto, debes estar conectado a una Wi-Fi distinta a la Wi-Fi del dron que disponga de conexión. Estas opciones tienen poco que ver con la comunicación entre el dron y el cliente por lo tanto no vamos a prestarlas demasiada atención.

En cambio, tanto para poder manejar el dron como para actualizar el firmware del dron, es imprescindible estar conectado a la red Wi-Fi del dron.

B.1 Manejando el dron

Para controlar el dron, accedemos a la opción **Take Off**. Nos aparecerá el siguiente menú.

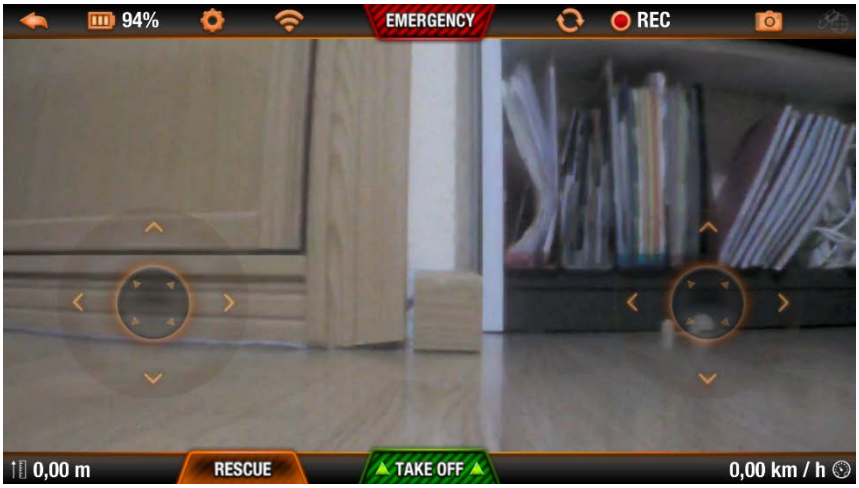


Figura B.3: Pantalla de manejo del dron

Pulsamos sobre "**Take Off**" para empezar a volar el dron. Se controla con los joysticks que vemos en la pantalla (aunque hay distintas opciones de manejo). Cuando nos cansemos de volarlo, pulsamos sobre "**Landing**" para que aterrice.

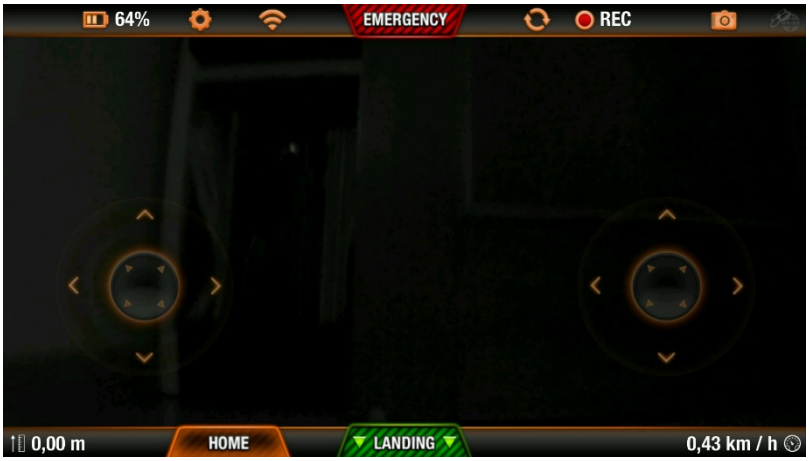


Figura B.4: Aterrizar dron

Si pulsamos sobre el **engranaje**, nos llevará al menú de opciones. Ahí podremos cambiar algunos parámetros de configuración del dron.

Figura B.5: Configuración

B.2 Emparejamiento

El **emparejamiento** podría considerarse como una opción que sirve para en cierto modo, aumentar la seguridad de las comunicaciones. Cuando esta opción está habilitada, el dron memoriza la MAC del dispositivo, y **todo tráfico que no venga de esa MAC será filtrado** (salvo los paquetes **ICMP**, y aquel tráfico que vaya al puerto **21** y **2049**). Esta opción se habilita desde el menú de opciones.

Figura B.6: Activación de emparejamiento

B.3 Visualizar nuestras fotos y vídeos

En el apartado Fotos/Videos, podremos visualizar las fotos y vídeos que hemos grabado.

Figura B.7: Fotos y vídeos desde la aplicación AR FreeFlight

B.4 Actualización del dron

El programa AR.FreeFlight tiene ya descargado el último firmware disponible. Comprueba la versión del firmware del dron, y si es anterior a la del firmware que tiene el dron, se conecta al dron e inicia el proceso de actualización.

C Configurar IP estática

Tras estar activada la opción de emparejamiento, tratamos de conectarnos a la red Wi-Fi **ardrone2**. El ordenador intentará conectarse, pero no podrá. Vamos a la opción "**Editar las conexiones**":

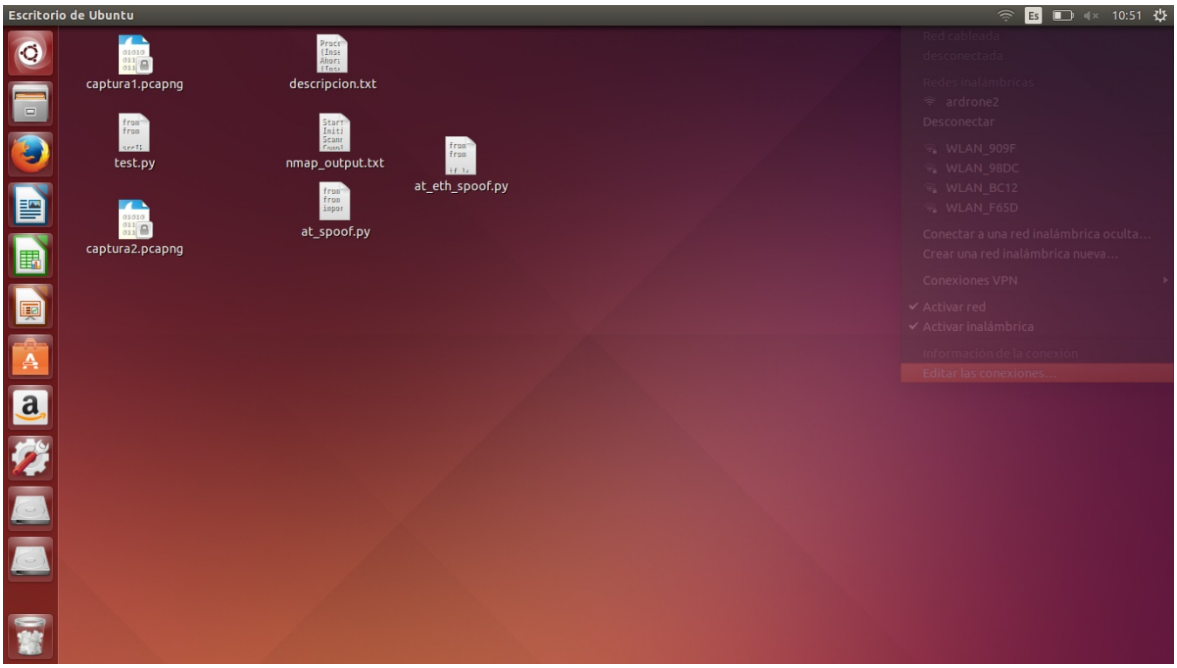


Figura C.1: Editar conexiones

Seleccionamos la red **ardrone2** y le damos a editar.

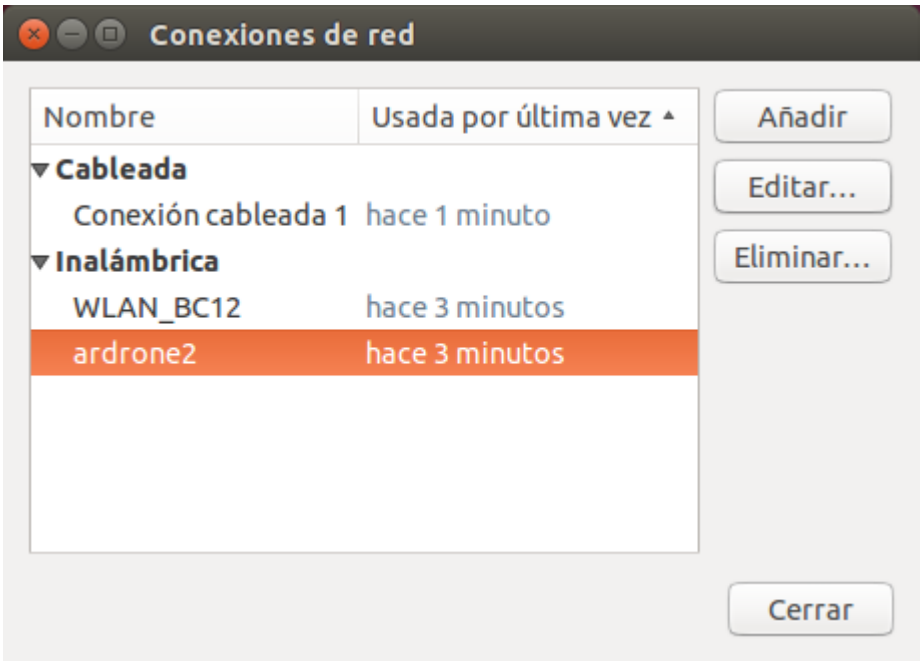


Figura C.2: Conexiones de red

Ahora, en **Ajustes de IPv4**, especificamos los datos tal y como se hace en la siguiente imagen:

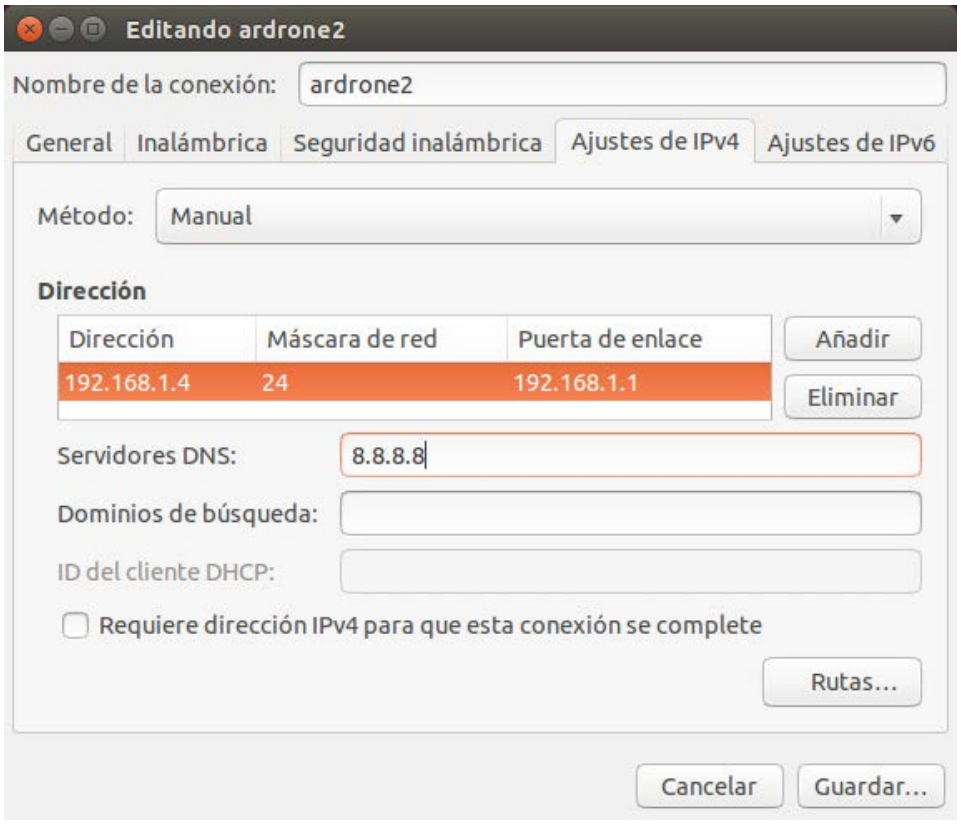


Figura C.3: Configurar IP estática

A continuación le damos a guardar, y podremos ver como al cabo de un tiempo nuestra máquina se ha conectado a la red Wi-Fi ardrone2 con éxito.

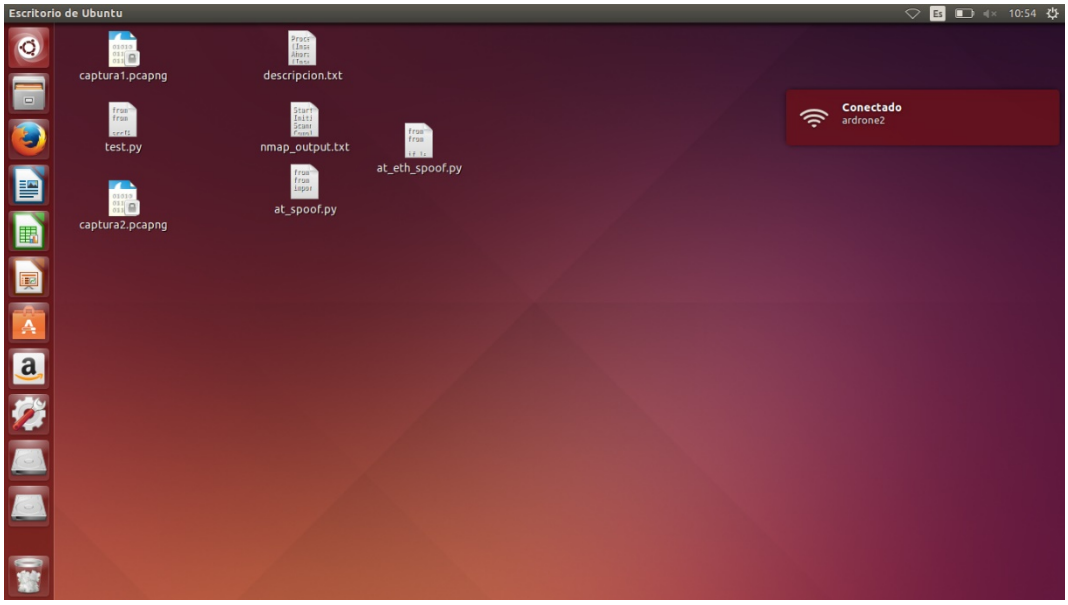


Figura C.4: Conexión realizada con éxito

D Explotando el FTP

Pudiendo acceder al dispositivo por medio de FTP, las posibilidades a la hora de realizar ataques son infinitas. En este anexo se describen aquellos ataques que un atacante podría estar interesado en realizar en la vida real: **robar uno o más archivos confidenciales**, **modificar algún archivo de suma importancia**, e **introducir un archivo malicioso**. Durante la explicación de estos ataques se conectará un dispositivo USB al dron, el cual contendrá una serie de archivos que en un caso práctico real pueden interesar a un atacante.

D.1 Robar uno o más archivos confidenciales

El primer ataque que a uno se le puede ocurrir realizar dada la situación descrita anteriormente es explorar el dispositivo USB en busca de información confidencial. Para realizar este ataque, nos conectamos al dron por FTP, y cambiamos de directorio al dispositivo USB.

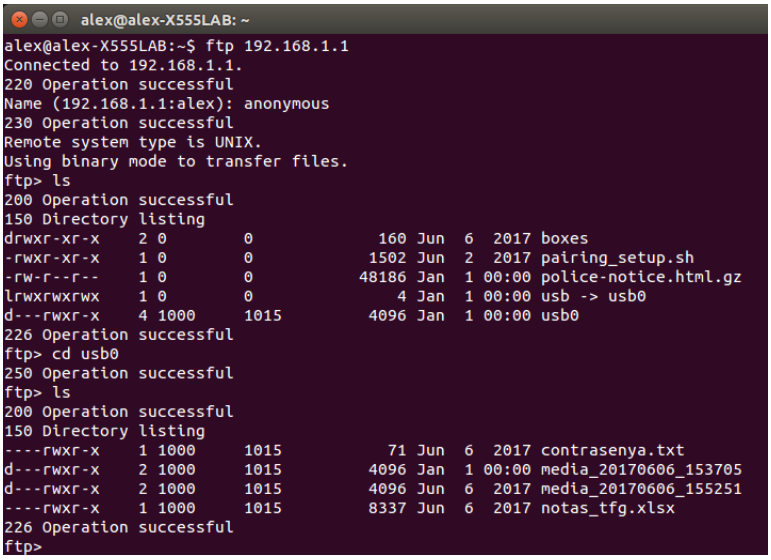


Figura D.1: Acceso a dispositivo USB desde FTP

Podemos ver que en el dispositivo USB hay un archivo contrasenya.txt. Descargamos ese archivo mediante el comando **get**.

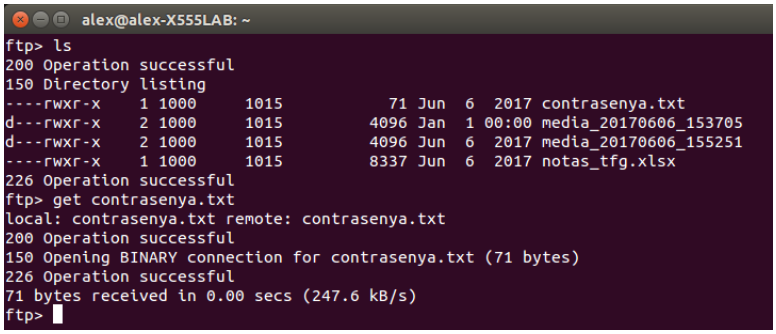


Figura D.2: Descarga del archivo confidencial “contrasenya.txt”

Una vez hemos descargado el archivo, lo abrimos, y vemos que contiene la contraseña de Moodle del dueño del dispositivo USB.

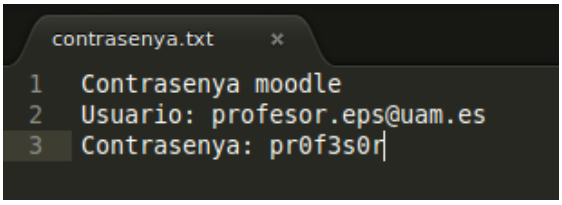


Figura D.3: Contenido del archivo “contrasenya.txt”

Este ataque, al igual que los siguientes, simplemente es una representación de un caso práctico bastante común.

D.2 Modificar algún archivo de suma importancia

Hemos visto que, a parte del archivo contrasenya.txt, en el dispositivo USB hay un archivo **notas_tfg.xlsx**. Nos descargamos ese archivo utilizando el comando **get**:

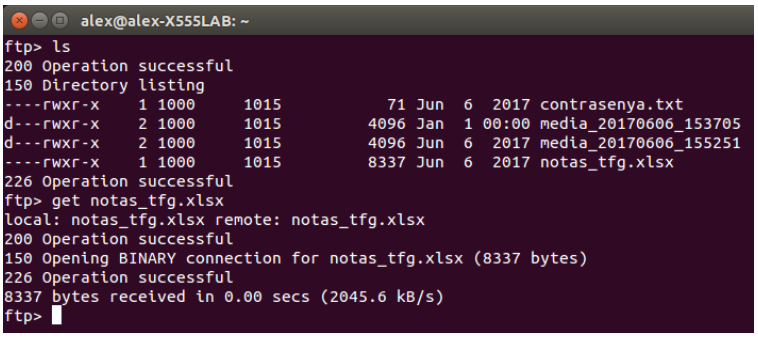
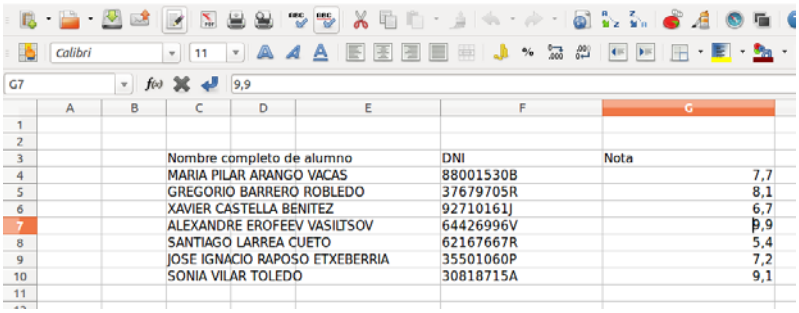


Figura D.4: Descarga del archivo confidencial notas_tfg.xlsx

Ahora abrimos el archivo y vemos su contenido. Modificamos nuestra nota, y nos ponemos un 9,9 (un 10 levantaría sospechas).



	A	B	C	D	E	F	G
1							
2							
3			Nombre completo de alumno			DNI	Nota
4			MARIA PILAR ARANGO VACAS			88001530B	7,7
5			GREGORIO BARRERO ROBLEDO			37679705R	8,1
6			XAVIER CASTELLA BENITEZ			92710161J	6,7
7			ALEXANDRE EROFEEV VASILTSOV			64426996V	9,9
8			SANTIAGO LARREA CUETO			62167667R	5,4
9			JOSE IGNACIO RAPOSO ETXEBERRIA			35501060P	7,2
10			SONIA VILAR TOLEDO			30818715A	9,1
11							

Figura D.5: Modificación del contenido del archivo notas_tfg.xlsx

Una vez hemos modificado el archivo, volvemos a subirlo al directorio raíz con el comando **put**.

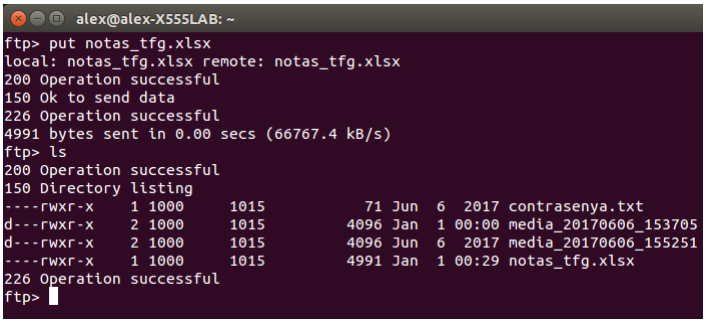


Figura D.6: Subida del archivo notas_tfg.xlsx

Acabamos de modificar un archivo confidencial sin que el dueño del dispositivo se dé cuenta.

D.3 Introducir un archivo malicioso

Otro ataque el cual un usuario estaría realizado en realizar sería introducir un archivo malicioso en el dispositivo USB, tratar de que el usuario de alguna manera lo ejecute, y de esta manera infectar su máquina.

Fijémonos atentamente en el formato de los archivos de vídeo:

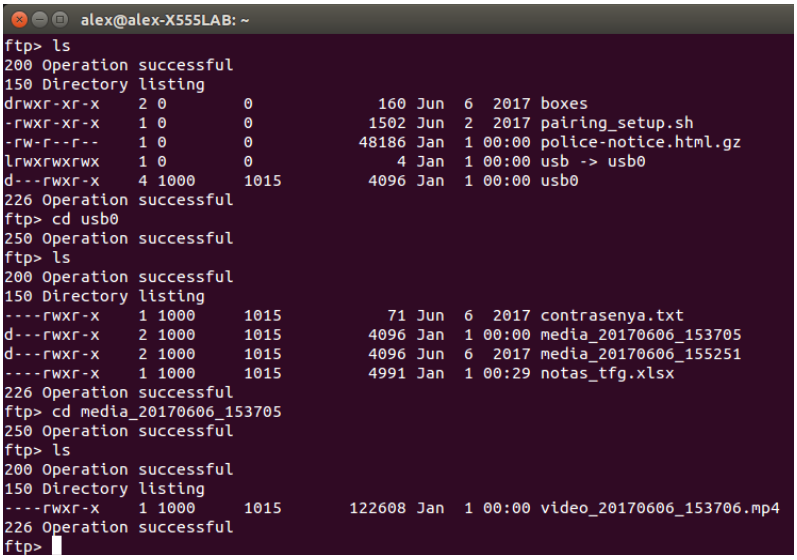


Figura D.7: Formato de los archivos de vídeo grabados por el dron

La forma más sencilla de hacer que el usuario caiga en la trampa consistiría en introducir un archivo malicioso que tenga un nombre con el mismo formato de nombre que los vídeos. Supongamos que el archivo troyano.exe que tenemos en nuestro sistema es un archivo malicioso, lo renombramos a un formato similar al que tiene un vídeo del dron:

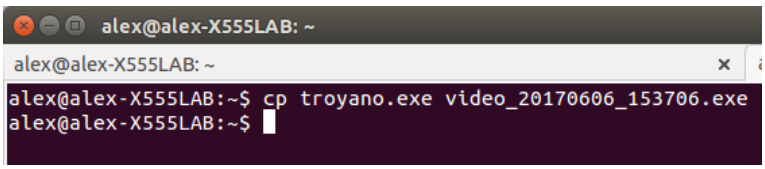


Figura D.8: Renombramiento de un archivo malicioso

A continuación subimos el archivo malicioso a la carpeta correspondiente y borramos el vídeo original.

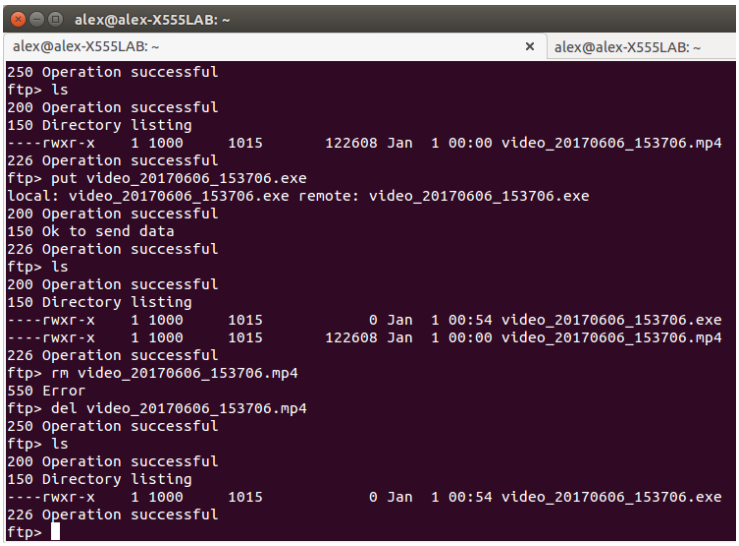


Figura D.9: Subida del archivo malicioso

Ahora solo nos queda esperar a que el usuario ejecute el archivo. Muchas veces los usuarios suelen tener desactivada la visualización de extensiones de archivos, por lo que si además nuestro archivo malicioso tiene un icono similar al de cualquier archivo de vídeo, tendremos el trabajo hecho.

Un ataque más elaborado podría consistir en añadir una macro al archivo .xlsx y que esa macro se encargase de ejecutar el archivo malicioso.

E Herramientas utilizadas

Para la realización de este trabajo se han utilizado las siguientes herramientas:

- Ordenador portátil con **Ubuntu 14.**04 instalado.
- El escáner de puertos **nmap**.
- La herramienta **arpspoof**, del paquete **dsniff**.
- El *sniffer* y analizador de protocolos **Wireshark**.
- La librería **Scapy**, sin la cual no sería posible desarrollar scripts en Python capaces de enviar y recibir paquetes a través de la red.

F Scripts

Los siguientes scripts se han utilizado para realizar los distintos ataques:

- **At_spoof.py** [26]. Este script envía paquetes al dron con la orden de aterrizar con la dirección IP origen suplantada. La dirección IP origen se indica como parámetro. El formato de ejecución es el siguiente:

  sudo python at_spoof.py ip_dispositivo

- **At_eth_spoof.py** [27]. Similar al anterior, pero además suplanta la dirección MAC origen. El formato de ejecución es el siguiente:

  sudo python at_spoof.py ip_dispositivo MAC_dispositivo MAC_dron

- **Deactivate_mac_filter.py** [28]. Este script esnifa paquetes hasta encontrar las claves necesarias para desactivar el emparejamiento. Una vez las ha encontrado, lo desactiva. El formato de ejecución es el siguiente:

  sudo python deactivate_mac_filter.py ip_dispositivo MAC_dispositivo MAC_dron

- **Infection.sh** [32]. Este script está a la espera de que se conecte algún dispositivo USB al dron, y en cuanto un dispositivo se conecta, copia un archivo malicioso en ese dispositivo, imitando las formas del dron a la hora de guardar los vídeos grabados.

G Vídeos

Con el fin de mostrar de manera más gráfica algunos ataques, se han realizado dos vídeos:

- El **primer vídeo** [25] muestra cómo podemos suplantar órdenes de manejo del dron estando el *pairing* desactivado.
- El **segundo vídeo** [29] muestra cómo podemos hacerlo si el *pairing* está activado, y también muestra cómo podemos desactivar el *pairing*.